RADC-TR-82-189
Final Technical Report
June 1982

RADC TESTABILITY NOTEBOOK

AD A118881

Hughes Aircraft Company

J. Byron
L. Deight
G. Stratton

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC
ELECTE
SEP 3 1982
S A D

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

82 09 03 002

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-82-189 has been reviewed and is approved for publication.

APPROVED: James Saporito

JAMES SAPORITO
Project Engineer

APPROVED: David C. Luke

DAVID C. LUKE, Colonel, USAF
Chief, Reliability & Compatibility Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
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-82-189	2. GOVT ACCESSION NO. A118881	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RADC TESTABILITY NOTEBOOK		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report Jan 80 - Apr 81
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) J. Byron L. Deight G. Stratton		8. CONTRACT OR GRANT NUMBER(s) F30602-80-C-0058
9. PERFORMING ORGANIZATION NAME AND ADDRESS Hughes Aircraft Company Engineering Services & Support Division PO Box 3310, Fullerton CA 92634		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 23380239
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (RBET) Griffiss AFB NY 13441		12. REPORT DATE June 1982
		13. NUMBER OF PAGES 388
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: James Saporito (RBET)

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

Inherent Testability
Comprehensive Testability
Supportability
Life Cycle Cost
Testability Design Techniques

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

Testability represents the inherent suceptibility of electronics systems to undergo valid functional performance testing, fault detection and fault isolation, together with the characteristics of the test system mix of Built-in-Test and external test equipments. A high degree of testability is necessary to achieve the highest levels of system performance and availability at least costs of system development and operation since good testability allows for discovery and correction of faulty system operation

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) with the least expenditure of program resources and within acceptable limits of elapsed time. The inherent testability needs to be systematically developed and integrated with the design of the system and the requirements for testability must be accorded the same level of recognition as performance, reliability, maintainability, availability, supportability and safety. The Testability Notebook provides fundamental guidance for systematic establishment throughout the development cycle of the requisite inherent testability and comprehensive testability of the test resource mix in combination with the prime system design.

PREFACE

The growth of the electronics technology has created challenges to the use of electronics systems in several respects. Not the least of these challenges is represented by the complexity of testing the systems in the field to determine functional status and to permit efficient fault detection and fault isolation.

The Rome Air Development Center (RADC) through its Testability Research Program has provided Government and industry with improved testability design tools and software. The Testability Notebook is an RADC-sponsored effort to define and synopsize a systematic methodology for including a high degree of testability in the evolving development of electronics systems.

The RADC Testability Notebook consists of an ordered and related collection of tasks oriented towards the evolution of prime system testability from concept formulation through deployed operations. The primary objective is to provide for achievement of cost-effective and efficient field testability of electronic systems.

It is hoped that this Notebook will be the first milestone in the rapid implementation of a simplified methodology for achieving the desired levels of testability and that high visibility may be accorded to the resultant benefits of higher operational readiness at reduced cost.

Mr. James Saporito was the RADC Project Engineer on this effort from its inception until his retirement. Requests for information concerning current and future additions to the Testability Notebook and other comments should be directed to:

Jerome Klion
RADC/RBET
Griffiss AFB NY 13441
AC 315-330-4726
AV: 587-4726

TABLE OF CONTENTS

Compendia Sections*

SECTION I CONCEPTUAL PHASE TASKS

SECTION II VALIDATION PHASE TASKS

SECTION III FULL SCALE DEVELOPMENT PHASE TASKS

SECTION IV PRODUCTION PHASE TASKS

SECTION V OPERATION AND SUPPORT PHASE TASKS

* A detailed Table of Contents (with Lists of Illustrations and Tables) precedes each Compendia Section.

LIST OF ILLUSTRATIONS

LIST OF TABLES

RADC TESTABILITY NOTEBOOK

INTRODUCTION

1.1 PURPOSE

The RADC Testability Notebook presents a consolidation of information presently available relating to testability design techniques, procedures, cost tradeoff tools, and the relationship of testability to other design disciplines and requirements.

1.2 STATEMENT OF PROBLEM

In the past testability design considerations have comprised weak links in the chain from system/equipment specification, through design, to the definition of built-in-test (BIT) and external test equipment (ETE) support hardware and software. Testability of the prime equipment has generally become a concern only during the AGE or GSE design effort which normally has occurred late in the development phase, when the operational hardware design is firm; changes to improve testability are thereupon assigned low priority because of associated schedule delays and higher costs. The resultant effect is interaction into operational use of electronic equipments and systems, which have high mean times to repair, high manpower rates (maintenance manhours per operating hour), retest OK (RETOK), could not duplicate (CND), and false alarm rates. These, in turn, reduce system availability and operational readiness and increase life cycle costs.

System/Equipment testability can be improved by treating testability as a design discipline starting in the conceptual phase and continuing through the acquisition process. Such a discipline when applied by management elevates testability to design factor stature as a peer to reliability, maintainability, availability and supportability. The nature, makeup and organization of that discipline, as it should be applied to each phase of the equipment/system acquisition process is contained in this Notebook.

1.3 Definition of Testability Functions, Tasks, and Interfaces

The Notebook takes cognizance of established system development phasing as the baseline or driving set of events for its structure. The five program phases were taken as Conceptual, Validation, Full Scale Development, Production and Deployment/Operation. System development objectives for each phase were analyzed and corresponding testability postures and objectives were then established for each phase, and a set of testability functions and tasks were derived.

The Notebook in addition treats two broad interface areas within a testability program. First, the set of relationships among the testability tasks within a program phase and from one phase to another. The second is the interface area between testability and its related disciplines including system and unit design, developmental test, maintainability, reliability, logistics and supportability. Program management makeup and guidance are provided for these as well as for the primary testability activities and tasks.

1.4 TECHNICAL RESULTS

The results of the Testability Notebook are documented in the detailed testability program flow charts, testability task compendia, and, application notes which forms its body.

1.5 Testability Program Flow

The flow charts indicate the functions, tasks, and interfaces, their order and logic applicable to each program phase. The flow charts in addition indicate the prerequisite activities necessary for the accomplishment of any set of functions and tasks. Figure 2-1 (page 2-14) presents the baseline overall task flow. Each phase is broken into functional activity areas, each of which encloses the tasks of that activity. A separate section of the compendia treats the tasks of each phase and includes a flow chart for the phase.

1.6. Testability Task Compendia

Each flow chart references a different set of testability task compendia which are coded by phase and function. The compendia in the Notebook are organized with respect to program phase and each phase is broken down into its relevant testability functions. Each function is then further broken down into its constituent tasks and activities. Technical guidance is then provided for each task/activity.

1.6.1 Testability Functions

The baseline set of testability functions is summarized as follows. Each function is developed into one or more tasks and presented in the Testability Notebook.

Conceptual Phase

- Establish the qualitative and quantitative testability requirements for the prime system.
- Conduct preliminary tradeoffs to establish the test system definition and to provide design criteria for prime system compatibility with the test philosophy.
- Incorporate testability requirements into the system specification.

Validation Phase

- Provide a Testability Program Plan.

- Provide detailed guidance for incorporation of testability in the prime system design.
- Perform detailed tradeoffs of Built-in-Test and external test equipment mixes.
- Analyze the testability characteristics of the evolving design.
- Document the results of the testability analysis.
- Develop specification segments for test and testability elements.
- Support and participate in the Preliminary Design Review.

Full Scale Development Phase

- Analyze and document detailed test requirements.
- Predict effectiveness of the defined test system.
- Document testability costs and benefits.
- Support and participate in the Critical Design Review.
- Plan and conduct a Testability Demonstration.
- Document a final testability analysis.
- Monitor developmental test.
- Conduct a pre-production testability readiness review.

Production Phase

- Monitor production and participate in change proposal activity.

Operations/Support Phase

- Monitor and evaluate field testability data.

2.0 TESTABILITY NOTEBOOK - SUMMARY

2.1 Technical Description of the Testability Problem

Testability is a significant key to achieving system performance and cost-effectiveness goals. A systematic approach is needed to establish and meet testability requirements beginning in the earliest program phases.

2.1.1 The Requirement for a Disciplined Approach to Testability Design

Increasing recognition is being given to correlations between system life cycle costs and the systems' testability characteristics. Reliability, maintainability, availability and producibility, among other disciplines, are essentially peers of testability, yet in modern program developments they are frequently treated much more formally than testability. Steps must therefore be taken to elevate testability to a higher status, with the objective of realizing significant long term cost benefits.

2.1.1.1 Testability Definition: Testability may be regarded as the inherent ability of an item to undergo valid, dependable functional testing and associated fault detection/isolation, within constraints of elapsed time, complexity of access, support equipment and functional procedures, and within set limits of manpower, material, and other resources. The tested item may be any level of indenture of a system of prime mission equipment or of some tangible element of mission support. Table 2-1 expresses this concept.

2.1.1.2 Testability Objectives: Functional test and condition monitoring are necessary to give assurance and expectation of mission success preparatory to or during operation, and in the course of maintenance or repair. Malfunction detection is necessary to permit consideration of alternative modes of operation and degree of mission success to be expected from use of each alternative mode. Annunciation of the malfunction is a prerequisite to making decisions to conduct maintenance and aids in determining whether or not maintenance will take place with or without system shutdown. Isolation of malfunctions is in turn a prerequisite to effecting repairs or otherwise restoring degraded components to required levels of operating performance.

2.1.1.3 Testability Approaches: Table 2-2 summarizes testability approaches. Testability characteristics must be injected early into designs. Poor testability conditions constitute design flaws and will lead to expensive design change procedures if not recognized before the design baseline is established. This need for identification and resolution applies from the inception of conceptual studies, through validation, engineering development, full scale development, production and the deployment phases. Testability is subordinate to performance, but system performance, to be effective, efficient, and economical, requires testability in design.

2.1.1.4 Testability in The System Heirarchy: Faults may be apparent at any point in the system heirarchy beginning with the initial fabrication or forming of single components. Testability therefore needs to be considered at the various stages of fabrication and assembly, and at the various steps of integrating assemblies into LRUs, LRU's into units, units into equipments, and equipment into systems. Furthermore, testability is a feature needed to ease and simplify acceptance and demonstration, to monitor deployed performance, as well as to detect, locate and isolate faults.

2.1.1.5 Testability Economics and Impacts. Properly managed, effective testability in design may be achieved without adding significant costs. Lack of effective testability incurs expense of much greater magnitude than the cost of achieving testability in design. Observe the MTTR term in the classical formulation of inherent availability,

$$A_i = (MTBF)/(MTBF + MTTR).$$

Poor testability characteristics first of all cause extended MTTR, with the direct impact of lost availability. Further, extended MTTR clearly implies added expenditures of costly manpower and test resources.

TABLE 2-1 TESTABILITY CHARACTERISTICS

TESTABILITY ASPECT	SIGNIFICANCE
o Thoroughness and ease of Condition Monitoring	o Testing is essential to full system effectiveness
- Fault Detection	o Operators need to know the status of system operating modes with full assurance
- Fault Isolation	o Valid, accurate, unambiguous detection and isolation of faults are key to achieving maximum operational availability
- Functional Verification	o Functional test is necessary to verify adequacy of performance before and after maintenance
o Constraints of - Elapsed Time - Simplicity of access - Human resources - Test materials - All cost-generating elements	Testability discipline in all aspects has heavy influence on the costs of operating and supporting prime mission equipment systems

TABLE 2-2 TESTABILITY APPROACHES

TESTABILITY REQUIREMENTS	SIGNIFICANCE
Inject into earliest designs	o Inherent testability is embedded in hardware design o Late incorporation generates extra costs
Provide active representation in all program life cycle phases	o Testability goals are established and monitored o Testability program plan to achieve goals at minimum cost is developed, and maintained
Testability posture at end of of each phase must be set to enter the next phase	o Testability as a discipline is similar to reliability and availability in that it should be also considered in DSARC and like reviews
Apply testability to all hardware indenture levels	o Production requires bottom-up integration testing o Operational and maintenance requires top-down testing.
Apply at all maintenance levels	o Maximize availability o Minimize resource consumption

2.1.2 The Role of the Testability Notebook

The Testability Notebook concept is to coalesce and codify extant literature and knowledge into tasks and procedures organized by procurement program phase. The notebook provides a roadmap for defining and satisfying testability needs in each phase. Although the Testability discipline is not yet fully formalized, there is no general lack of testability information and data. Much information exists, varying from theoretical dissertations to detailed descriptions of methods applicable to certain elements. Analysis and consolidation of such data, together with opinions, techniques, and methods gathered in interviews has resulted in the Testability Notebook which codifies the information for ready use.

2.1.2.1 Testability Notebook Concept. Table 2-3 (page 2-4) illustrates, in summary form, the basic concept taken for the codification of tasks by development phase. Note that the testability tasks relative to each phase are addressed in terms of both technical and management practices required. Systematic, standardized, and consistent practices have been identified or derived as appropriate to each phase of the life cycle. The practices include procedures, tools, tradeoffs, and other methods making up the techniques, and the means employed to manage and control achievement of objectives as can be seen no new or unique system engineering or management concepts are required for an effective testability program. This same general pattern applies to testability as applies to other system engineering components.

TABLE 2-3 TESTABILITY NOTEBOOK CONCEPT

Testability Tasks	Task Performance Techniques	Management/Control Techniques
Conceptual Studies		
• Establish functional requirements • Establish goals • Outline program plan	• Analysis of system utilization & support concept • Analysis of performance requirements • Close interface with related disciplines	• Approve goals • Approve plan/program concept • Establish criteria for next phase entry
Validation Phase		
• Complete program plan • Inject testability into evolving design • Select test system • Plan ED/FSD activities in detail	• Extensive coordination with related disciplines and program management • Input to system specifications • Develop test subsystem specifications	• Approve plan • Include testability in design reviews • Approve specifications • Establish criteria for next phase entry
Engineering/Full-Scale Development Phase		
• Develop test element detail specifications • Develop test system elements • Train human resources • Demonstrate goal achievement	• Growth of prior stage activities • Parallels prime mission equipment techniques • Operate proofing/testing plans	• Approve specifications • Witness and approve demonstrations • Establish criteria for next phase entry
Production Phase		
• Produce test system elements • Deliver and deploy elements • Assure smooth transition to service use • Institute information feedback	• Parallels prime mission equipment production/delivery/initiation techniques • Monitor initial use • Assist users	• Include testability indicators in monitoring initial operational effectiveness
Deployment/Operations Phase		
• Monitor performance • Improve test system • Control changes	• Collect/analyze operational data • Sustain representation in configuration control	• Use operation and utilization standards

TABLE 2-2 TESTABILITY APPROACHES

TESTABILITY REQUIREMENTS	SIGNIFICANCE
Inject into earliest designs	o Inherent testability is embedded in hardware design o Late incorporation generates extra costs
Provide active representation in all program life cycle phases	o Testability goals are established and monitored o Testability program plan to achieve goals at minimum cost is developed, and maintained
Testability posture at end of of each phase must be set to enter the next phase	o Testability as a discipline is similar to reliability and availability in that it should be also considered in DSARC and like reviews
Apply testability to all hardware indenture levels	o Production requires bottom-up integration testing o Operational and maintenance requires top-down testing.
Apply at all maintenance levels	o Maximize availability o Minimize resource consumption

2.1.2 The Role of the Testability Notebook

The Testability Notebook concept is to coalesce and codify extant literature and knowledge into tasks and procedures organized by procurement program phase. The notebook provides a roadmap for defining and satisfying testability needs in each phase. Although the Testability discipline is not yet fully formalized, there is no general lack of testability information and data. Much information exists, varying from theoretical dissertations to detailed descriptions of methods applicable to certain elements. Analysis and consolidation of such data, together with opinions, techniques, and methods gathered in interviews has resulted in the Testability Notebook which codifies the information for ready use.

2.1.2.1 Testability Notebook Concept. Table 2-3 (page 2-4) illustrates, in summary form, the basic concept taken for the codification of tasks by development phase. Note that the testability tasks relative to each phase are addressed in terms of both technical and management practices required. Systematic, standardized, and consistent practices have been identified or derived as appropriate to each phase of the life cycle. The practices include procedures, tools, tradeoffs, and other methods making up the techniques, and the means employed to manage and control achievement of objectives as can be seen no new or unique system engineering or management concepts are required for an effective testability program. This same general pattern applies to testability as applies to other system engineering components.

TABLE 2-3 TESTABILITY NOTEBOOK CONCEPT

Testability Tasks	Task Performance Techniques	Management/Control Techniques
Conceptual Studies ● Establish functional requirements ● Establish goals ● Outline program plan	● Analysis of system utilization & support concept ● Analysis of performance requirements ● Close interface with related disciplines	● Approve goals ● Approve plan/program concept ● Establish criteria for next phase entry
Validation Phase ● Complete program plan ● Inject testability into evolving design ● Select test system ● Plan ED/FSD activities in detail	● Extensive coordination with related disciplines and program management ● Input to system specifications ● Develop test subsystem specifications	● Approve plan ● Include testability in design reviews ● Approve specifications ● Establish criteria for next phase entry
Engineering/Full-Scale Development Phase ● Develop test element detail specifications ● Develop test system elements ● Train human resources ● Demonstrate goal achievement	● Growth of prior stage activities ● Parallels prime mission equipment techniques ● Operate proofing/testing plans	● Approve specifications ● Witness and approve demonstrations ● Establish criteria for next phase entry
Production Phase ● Produce test system elements ● Deliver and deploy elements ● Assure smooth transition to service use ● Institute information feedback	● Parallels prime mission equipment production/delivery/initiation techniques ● Monitor initial use ● Assist users	● Include testability indicators in monitoring initial operational effectiveness
Deployment/Operations Phase ● Monitor performance ● Improve test system ● Control changes	● Collect/analyze operational data ● Sustain representation in configuration control	● Use operation and utilization standards

2.1.2.2 Existing Guides to Testability Techniques. The testability tasks of each phase can be defined by consideration of well-known and documented test guides as is done for maintainability and reliability programs via handbooks, manuals, and pamphlets generated by various service agencies. Such as:

(1) RADC TR-79-327 "An Objective Printed Circuit Board Testability Design Rating System"
(2) RADC-TR-79-309 "BIT/External Test Figures of Merit and Demonstration Techniques"
(3) TR-3826 "A Framework for Designing Testability into Electronic Systems", Naval Surface Weapons Center
(4) ASD-TR-79-5013 "BIT/SIT Improvement Project"

Extracts from and reference to applicable documents are contained in the Notebook content.

2.1.2.3 Testability Application and Development in Program Phases. During each program phase different but related tasks of appropriate scope have to be addressed. As a particular example, in the conceptual studies phase, a preferred method of establishing testability goals might be documented. A related interface with reliability assessment could also be identified. Sub-allocations of times-to-repair (in establishing the maintainability goal or specified value of MTTR) might be made to levels of test. The availability discipline may be interfaced to aid in establishing test concepts for scheduled and unscheduled maintenance. Carrying this example to the full scale development phase, standardized procedures could be documented to measure the degree of achievement of test times and allocations. In the deployment/operations phase, standardization can be extended to collection of field data independently or in conjunction with AFM 66-1-based programs, or similar programs of Army and Navy Services.

2.2 Definition of Life Cycle Testability Functions and Tasks Making up the Notebook

As indicated previously, established system development phasing was taken as the baseline or driving set of events to which all supporting disciplines are to be keyed. The five program phases for grouping events were taken as Conceptual, Validation, Full Scale Development, Production and Deployment/Operation.

2.2.1 A Simplified Version of the Baseline Phase Arrangement of Testability Functions, Tasks, and Interfaces

In order to provide an overview of the content of the phased arrangement of testability functions, tasks and interfaces, the following simplified baseline for the organization of a testability program over the five program phases is provided. The compendia which follow this section provide detailed engineering and management guidance to the testability acquisition process for and during each phase.

2.2.1.1 Testability Program Activities in the Conceptual Phase. The basic conceptual phase program activities are to conduct system feasibility studies, including identification of alternatives; to establish technical, military, and economic bases for acquisition and to decide whether or not to pursue the program. It is necessary to consider testability concepts in this phase because of the weight their consideration contributes to the decision process, and to overall program costs. Table 2-4 summarizes fundamental testability factors that most appropriately should be accounted for during the conceptual phase. The testability relation to other disciplines is indicated in the summary data in Table 2-5.

TABLE 2-4 TESTABILITY OPPORTUNITIES IN THE CONCEPT PHASE

- Establish Testing Concept
- Outline a Testability Program
- Define functional testing requirements
  - BIT/BITE versus external test equipment
  - Test concepts at hardware indenture levels (match existing hardware concept)
  - Test concepts at maintenance levels
    - Organizational
    - Intermediate
    - Depot
- Establish qualitative/quantitative testing goals
  - Thoroughness of condition monitoring
  - Time to detect (isolate)
  - Time to complete functional test
  - Man hours allocation
  - Cost allocation
  - Management exception trigger level
  - Testability figure of merit/achievement goals/thresholds

TABLE 2-5 CONCEPTUAL PHASE TESTABILITY INTERFACES WITH OTHER DISCIPLINES

Discipline	Testability Relation
● Reliability	- Failure rates - test rates allocations - Critical areas - thoroughness of test
● Maintainability	- Allocation of access to test points - MTTR allocations - Functional design for maintainability - Functional design for testability
● Design	- Accommodation of reliability/maintainability/ testability requirements - Design/BIT ratio allocation
● Availability	- Test time allocation - Scheduled vs unscheduled testing concept
● Life Cycle Cost/ Design to Cost	- Costs of alternative testing approaches
● Supportability	- Compatibility of testing approach alternatives with logistic support practices of procuring/ using agency
● Management	- Recognition of testability as a program element - Need for management exception trigger levels and allocation

2.2.1.2 **Testability Activities in the Validation Phase.** The key testability activity of the Validation Phase is the creation of a detailed testability program plan. The plan sets clear criteria for the achievement of testability goals and objectives in the subsequent program phases. This activity supports the main program purpose of the validation phase which is to prepare the system development concept. This phase is the optimum point for the testability program outline to be filled in and refined by progressive, interactive techniques. The related system and testability activities during validation are outlined in Table 2-6.

These testability activities contribute to the system definition, particularly as requirements in the system specifications. An outgrowth of the testability program definition is the development of specifications for test systems, and a preliminary listing of test equipment and test resource requirements. As system design detail fills in, the BIT/BITE versus external test allocation is refined over deeper levels of indenture. Similarly, qualitative and quantitative testability measures and aims are more closely related to specific functional areas and element, and/or to specific indentures as each particular design situation warrants. This brings up the need for a study to determine the best and most economic methods and tools for testing assemblies of complex devices such as microprocessors, ROMS, RAMS, PROMS and other VLSI/VVLSI devices of current and foreseeable technologies.

TABLE 2-6 VALIDATION PHASE ACTIVITIES

- **Basic Program Activities**
    - Prepare System Development Concept
    - Define program objectives
    - Define program issues
    - Detect potential special problems in logistic or other support disciplines
    - Define performance and cost parameters
    - Assess program, cost schedule performance risks
    - Identify system alternatives
    - Develop strategy for system and support element acquisition
    - Perform breadboard testing and verification

- **Testability Activities**
    - Input testability requirements into system specifications
    - Develop a preliminary testability program plan
    - Develop preliminary testing system specifications
    - Refine BIT/BITE/external test allocations
    - Develop preliminary listing of test equipment and other test resource candidates
    - Allocate quantitative/qualitative goal factors to functional areas and elements
    - Define preliminary test procedural requirements

2.2.1.3 Relationship With Other Disciplines During the Validation Phase. The maintenance of direct interactions with other disciplines such as reliability, maintainability, and system design is essential to the development of a superior testability program. During the validation phase the opportunities for interaction between testability and other design disciplines increase rapidly in expanse and in depth of detail. Table 2-7 outlines these relationships and some of the more significant impact areas.

Of great importance at this stage is the interface with system and subsystem design engineers in the applications of complex, difficult-to-test technologies. Detailed planning during this phase ensures availability of the testing capabilities and facilities that will be required in the following development, production, deployment, and operations phases. Opportunities must be exploited at this stage to optimally allocate weight, space, and power to BIT, condition monitoring in general, and maintenance/test functions. In addition to the relationship between reliability and testability allocations, testability aspects must directly consider failure modes and effects and critical items.

Major interfacings occur between maintainability and testability because of their very close interdependence. In many respects both consider the same elements but in different aspects. Some measures of testability are also measures of maintenance actions, particularly time-to-fault-detect and time-to-fault-isolate. Availability is also directly related to both of these other disciplines because of the need for and consumption of, system time to perform some actions.

The logistics system and testability have a number of points of interface, the more important being availability of support facilities and equipment, life cycle and design-to-cost aspects, supportability characteristics, and the maintenance plan.

Definition and resolution of the interdisciplinary problems between testability and interfacing disciplines is a major objective of the testability notebook.

TABLE 2-7 VALIDATION PHASE DISCIPLINARY INTERFACES

Interfacing Discipline	Testability Concerns
● Reliability	- Failure rates allocation in greater depth - Failure modes and effects - Critical items
● Maintainability	- Maintenance measures -- MTTR -- MDT -- MMH -- Access Time - Accessibility - Physical and functional hardware design
● Design*	- Condition monitoring techniques - Allocation of circuitry to BIT
● Availability	- Time to test - Scheduled versus unscheduled test - Condition monitoring techniques
● Life Cycle Cost/ Design to Cost	- Test-related costs - Test-driven costs and alternatives
● Supportability	- Compatibility with logistic support planning and practices
● System Test	- Coordinate planning and feedback - Software requirements and cost allocations - Hardware interface requirements and cost allocations
● Program Management	- Recognition of need for testability - Refinement of management exception trigger levels

*The design interface also accommodates certain of the concerns listed with other interfacing disciplines.

2.2.1.4 Activities and Interfaces in the Engineering Development and Full Scale Development Phases. The main purpose of the engineering development/ full scale development phase is to design and develop the system, giving due consideration for maintainability, logistic, and testability factors. The related system and testability activities are outlined in Table 2-8. Testability activities in this phase consist primarily of proofing, polishing and refining the testability concept and testability program plan developed in the preceding phases. Feedback of information from applying the plan in test programs and demonstrations allows it to be fine-tuned to fulfill the requirements of the production and deployment phases.

Specific activity occurs in the development of detail specifications for test equipments and resources and their acquisition proof testing. Test procedures are developed and proofed using prototype and development model hardware, which in turn enables determination of the achievement of testability measurement goals. Particular refinements are being made during this phase in activities related to logistics support, calibration, and maintenance of test equipment, and to facilities. These activities are now extended from strict system application to consideration of system support facilities. Another refinement is the activity to demonstrate by measurements the achievement of assigned testability goals. The development of test procedures for specific items of hardware is the culmination of tasks associated with selection of test equipment, definition of test requirements, hardware design, and other test resources. These specific instances are reflections of the intensified activity to completely define the testability concept and the plan in all of its aspects.

The culmination of all this effort during this phase is a fully-refined and proofed testability program plan, firm testability concept, and a completely-ready test system. These three testability elements must be ready for active use in the final production, deployment, and operation phases.

TABLE 2-8 ENGINEERING DEVELOPMENT/FULL SCALE DEVELOPMENT ACTIVITIES

- Program Activities
  - Develop Unit Fabrication Specifications
  - Develop all Specifications in full detail
  - Design complete system
  - Design and test hardware (DTE/OTE)
  - Develop logistic support planning and elements
  - Demonstrate Achievement of
    - Performance parameters
    - System and support operational goals
    - System and support utilization goals
  - Initiate system configuration control
  - Develop modifications planning system
- Testability Activities
  - Develop specifications for test elements and resources
  - Develop and proof test elements and resources
  - Develop and validate test procedures
  - Develop logistics and calibration support for test elements
  - Develop and conduct cadre test and maintenance training
  - Develop configuration planning for test system, keep consistent test system (with PME)
  - Develop test software configuration control procedures
  - Demonstrate achievement of testability goals
  - Finalize planning for test in production, operator & deployment phases
  - Modify testability plan to reflect all changes

In the engineering development/full scale development phase the interactions, impacts and influences between testability and other design characteristics reach peak activity levels . Close coordination and cooperation by personal direct consultations is needed to keep all design elements moving ahead in phase with one another and with schedule milestones. All persons involved in the total system concept must be aware at all times of the essential purpose of their combined effort: the achievement of the system goals of operational readiness, effectiveness, and supportability on a production basis in the deployment/operational arena. Interface activities shown in Table 2-9 are mainly a further intensification of those in the validation phase, with the growth that is to be expected as the details of system definition are filled in. It is important in this phase that the testability engineer has a position on the configuration control board to ensure that testability features are not neglected when considering modifications, changes and redesigns. It is also during this phase that feedback from the test and evaluation portion of the development activities begins to emerge. Positive action will enable the most effective use of this data, particularly where results indicate that corrective action is needed. Hence, this information is an important key to smoothing and polishing the interdisciplinary problems. Closed loop problem solution procedures are required in all disciplines including testability.

TABLE 2-9 TESTABILITY INTERFACES IN ENGINEERING/FULL SCALE DEVELOPMENT

- Largely the same as in the validation phase but at greatest intensity, depth, detail, and personal contact levels, e.g.:
    - System Specification
    - BIT/BITE Test Allocation
    - Reliability
    - Maintainability
    - Availability
- Provide feedback from test and evaluation activities
- Establish a voice in configuration control

2.2.1.5 Activities and Interfaces in the Production/Deployment Operational Phases. In the production/deployment/operation phases, the prime mission equipment and its supporting elements are produced, positioned in using location, and placed into operational use. Since these phases normally overlap, sustaining support for fielded equipments gradually passes from the manufacturer to the acquiring/using agencies, until past the end of production when the user/maintainer has full responsibility for the sustaining effort.

Table 2-10 opposite lists the main testability activies and the discipline interface function in these phases. The major activities involve application and evaluation of the plan and its constituent elements. A substantial portion of this activity is directed toward improving performance by enforcement of standards and, where needed, procedural changes or improvements in supporting hardware, software, or facilities. Associated with this is the sustaining effort to ensure that testability requirements are given due consideration in all design changes, whether instituted for producibility purposes or to improve system performance. The testability engineer must retain his position on the change control board. Another significant testability task of these phases is that of evaluating lessons learned and their incorporation into new and ongoing testability programs.

TABLE 2-10 ACTIVITIES AND INTERFACES OF THE PRODUCTION AND DEPLOYMENT/OPERATIONS PHASES*

Program Activities

- Prime Mission Equipment System and supporting elements are produced and positioned for operational use.
- At close of production phase sustaining support gradually passes to Government acquiring, operating, and support agencies.
- Systems and supporting elements are utilized in operational activities.

Testability Activities

- Monitor testability element performance.
- Obtain feedback to compare performance to program standards and goals.
- Improve performance through enforcements and/or procedure changes and/or prime mission equipment/Support Hardware modifications.
- Improve performance through requisite maintenance/modification of test software.
- Incorporate lessons learned into new or in process testability programs.

Testability Interfaces

- Largely the same as in the Engineering Development/Full Scale Development Phase but at much reduced intensities and with much lessened personal contact, e.g.:
    - System Test Effectiveness
    - Personnel Skill Demands
    - Reliability/Maintainability Feedback
- Government using and supporting agencies

*NOTE: Activities of deployment/operations phase may be Government responsibility, depending upon contract provisions and the transition plan.

2.3 Guide for Using the Notebook

The results of the Testability Notebook Development are documented in testability program flow charts, testability task compendia, and application notes. The compendia (with flow charts), and the material described previously (including a master flow chart and a description of the compendium format) is contained in the sections which follow.

2.3.1 Testability Program Flow Charts

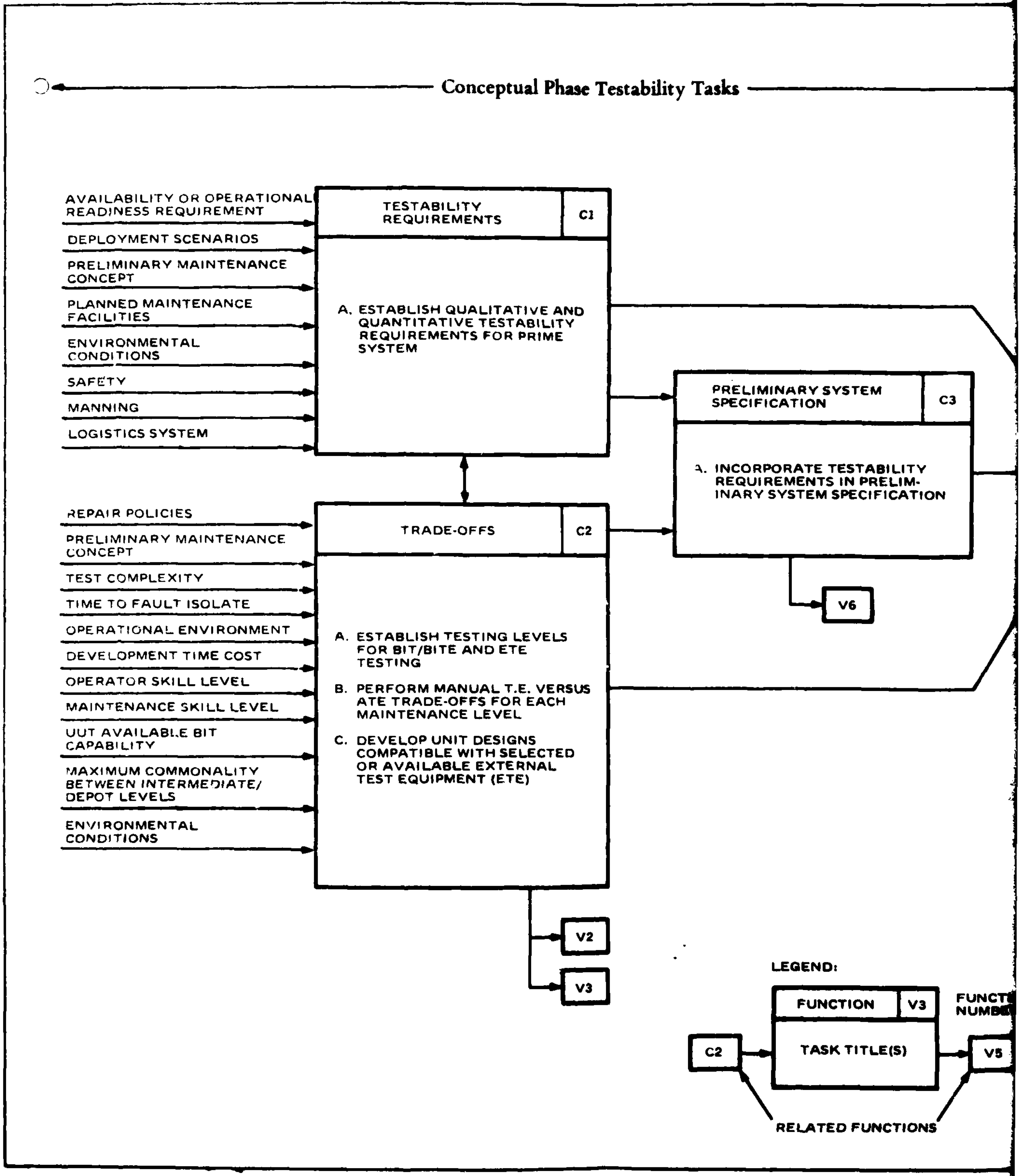
The flow charts serve as an index to the testability tasks and represent the overall philosophy of program flow derived from the analysis. A composite flow chart, covering all phases, is presented in Figure 2-1. Each phase is broken into functional activity areas, shown as numbered blocks; each block encloses the tasks of that activity. Relationships, interfaces, interactions and flows among functional activity areas and tasks are represented. The complex interactions between non-adjacent functions may be directly connected by arrows; or may be shown by small arrowed rectangles containing individual numerals which correspond to the connecting function. The flow charts provide the logical flow for the testability activities and tasks (and their necessary inputs) to be undertaken during each phase, and the relationships among such activities and tasks from one phase to the next. As such they provide end to end guidance as to when each activity or task is relevant and identifies their prerequisite inputs.

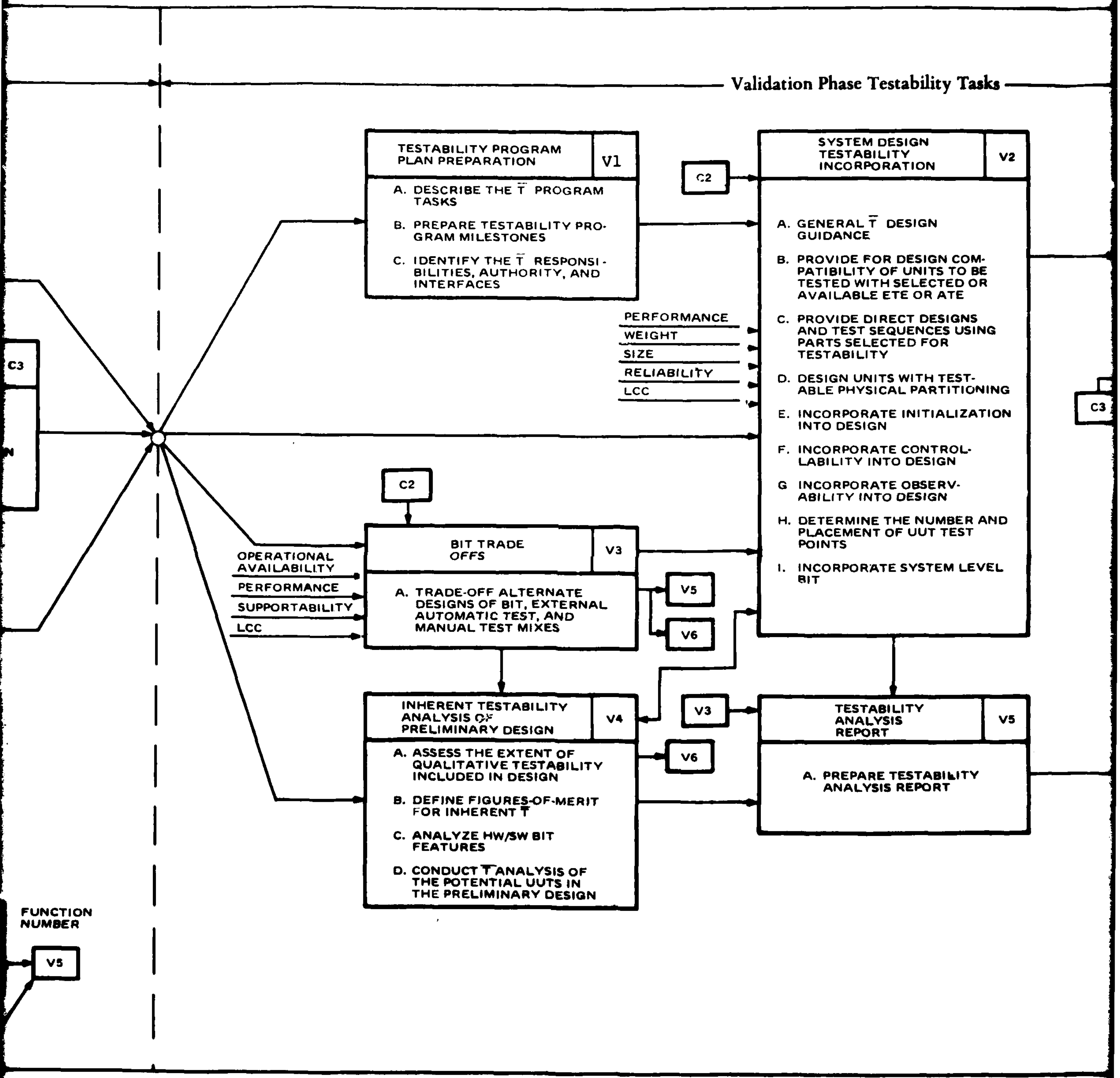
The flow for the conceptual phase illustrates the mechanization. Blocks C1, C2 and C3 are functional areas containing 1, 3 and 1 tasks respectively. Each task is identified by a title and a reference number. Unnumbered inputs to the blocks signify data accessed from outside the testability discipline. Results of function C2 are used in functions V2 and V3. Functions C1, C2 and C3 all connect to function V1 (the first sequential activity of the Validation phase).

In function V2 task V2A is artificial. Its purpose is to present a check list of design guidance appropriate to the general task of injecting testability into the design. (Features of the check list which are specifically appropriate to other tasks, e.g., controllability and observability, V2F and V2G, are repeated in those task compendia.)

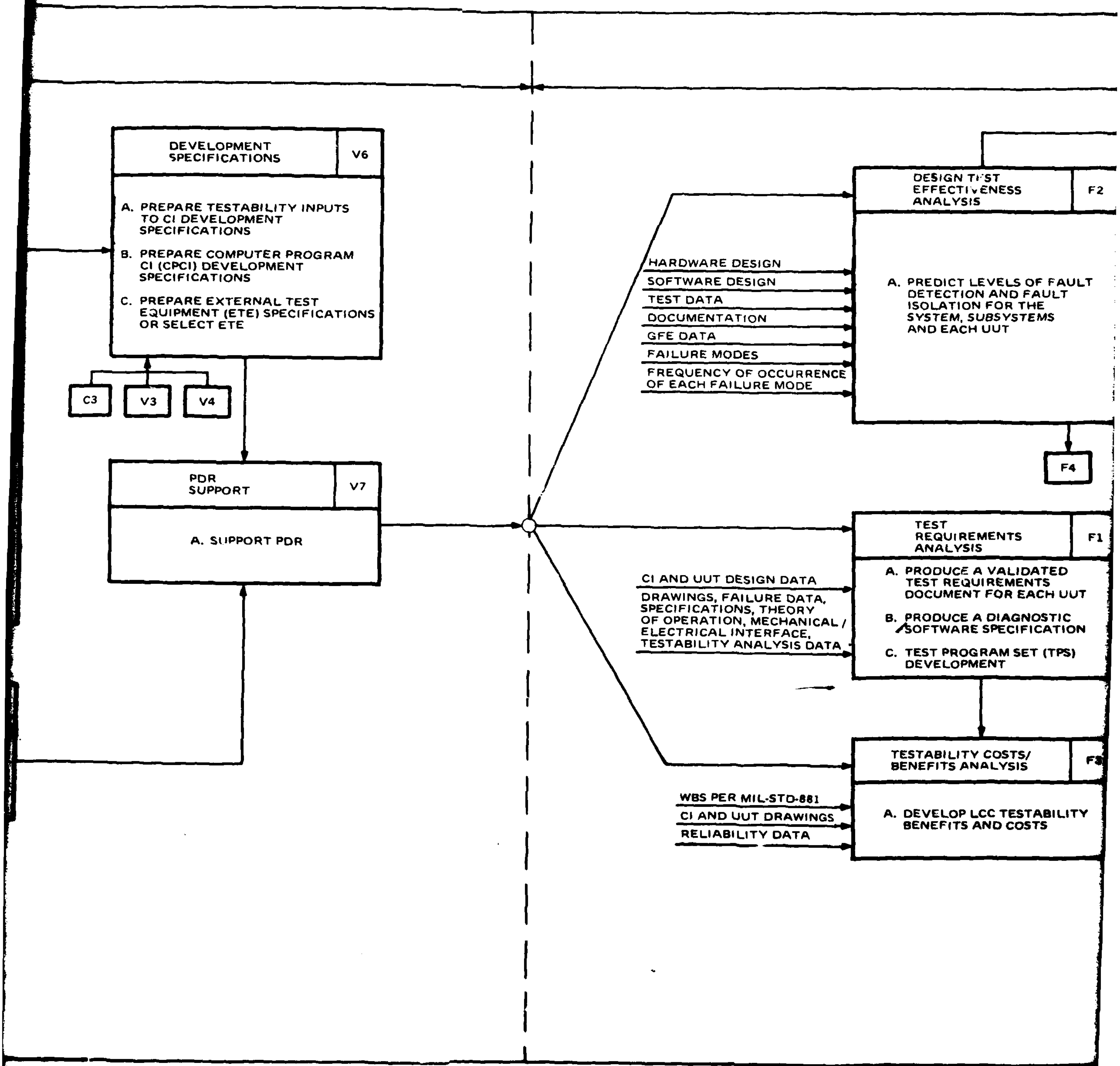
The placement of the Critical Design Review in the Full Scale Development flow exemplifies the generalization of the flow charts. In any particular system program, there will be preferential ways of phasing events which will supersede previous general guidance. The CDR might thus occur earlier or later in the phase, and the performance of any related task would need adjustment of its inputs and outputs to achieve the objectives. The arrangement of the tasks and task compendia is intended to support the needed flexibility.

In the Compendia Sections of the Notebook, the program flow is repeated in the form of an individual flow chart for each phase.


Conceptual Phase Testability Tasks

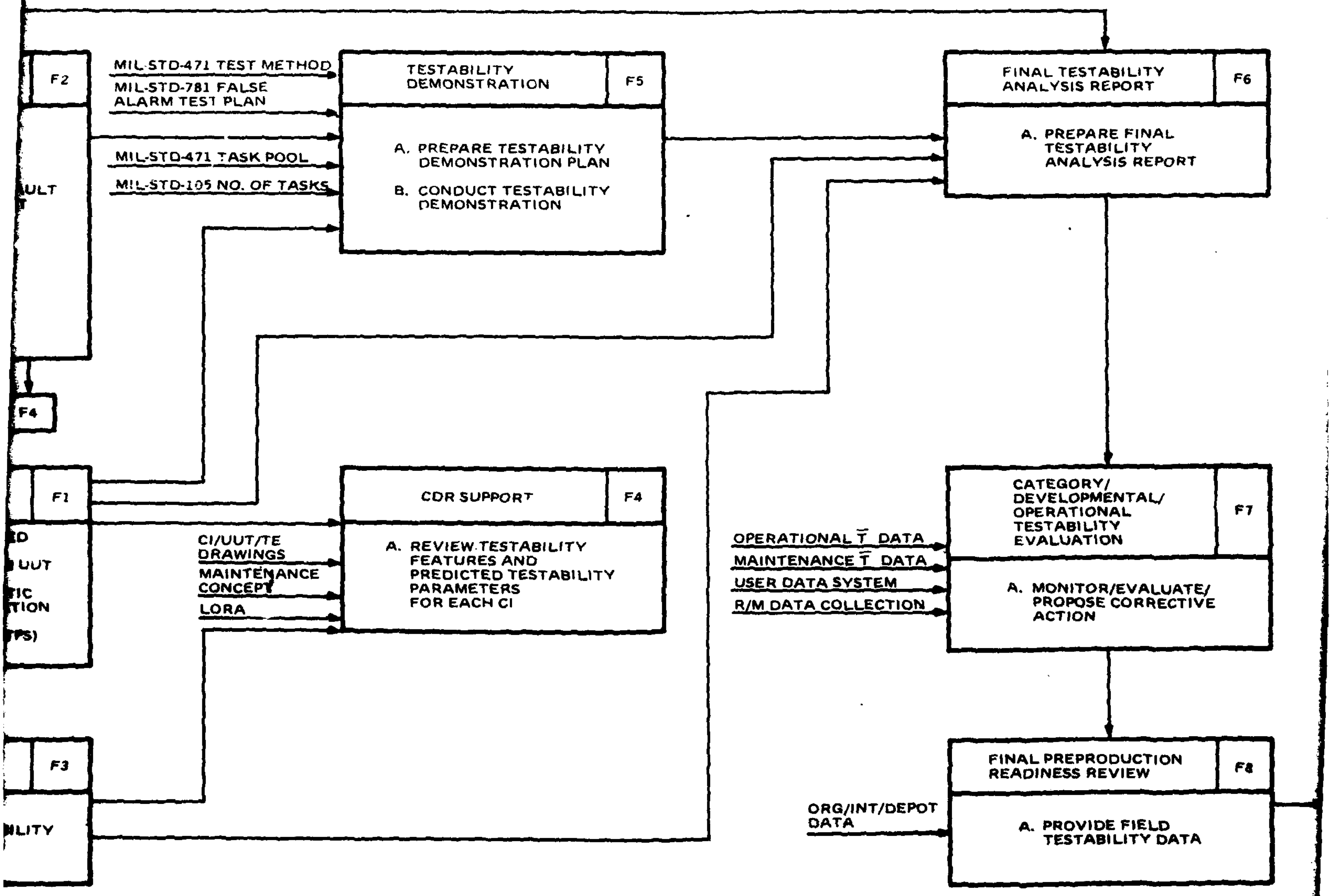

Validation Phase Testability Tasks
TESTABILITY PROGRAM PLAN PREPARATION
V1
A. DESCRIBE THE T̄ PROGRAM TASKS
B. PREPARE TESTABILITY PROGRAM MILESTONES
C. IDENTIFY THE T̄ RESPONSIBILITIES, AUTHORITY, AND INTERFACES
C2
SYSTEM DESIGN TESTABILITY INCORPORATION
V2
A. GENERAL T̄ DESIGN GUIDANCE
B. PROVIDE FOR DESIGN COMPATIBILITY OF UNITS TO BE TESTED WITH SELECTED OR AVAILABLE ETE OR ATE
C. PROVIDE DIRECT DESIGNS AND TEST SEQUENCES USING PARTS SELECTED FOR TESTABILITY
D. DESIGN UNITS WITH TESTABLE PHYSICAL PARTITIONING
E. INCORPORATE INITIALIZATION INTO DESIGN
F. INCORPORATE CONTROLLABILITY INTO DESIGN
G INCORPORATE OBSERVABILITY INTO DESIGN
H. DETERMINE THE NUMBER AND PLACEMENT OF UUT TEST POINTS
I. INCORPORATE SYSTEM LEVEL BIT
PERFORMANCE
WEIGHT
SIZE
RELIABILITY
LCC
C3
C3
N
C2
BIT TRADE OFFS
V3
A. TRADE-OFF ALTERNATE DESIGNS OF BIT, EXTERNAL AUTOMATIC TEST, AND MANUAL TEST MIXES
OPERATIONAL AVAILABILITY
PERFORMANCE
SUPPORTABILITY
LCC
V5
V6
INHERENT TESTABILITY ANALYSIS OF PRELIMINARY DESIGN
V4
A. ASSESS THE EXTENT OF QUALITATIVE TESTABILITY INCLUDED IN DESIGN
B. DEFINE FIGURES-OF-MERIT FOR INHERENT T̄
C. ANALYZE HW/SW BIT FEATURES
D. CONDUCT T̄ ANALYSIS OF THE POTENTIAL UUTS IN THE PRELIMINARY DESIGN
V6
V3
TESTABILITY ANALYSIS REPORT
V5
A. PREPARE TESTABILITY ANALYSIS REPORT
FUNCTION NUMBER
V5

2


DEVELOPMENT SPECIFICATIONS
V6
A. PREPARE TESTABILITY INPUTS TO CI DEVELOPMENT SPECIFICATIONS
B. PREPARE COMPUTER PROGRAM CI (CPCI) DEVELOPMENT SPECIFICATIONS
C. PREPARE EXTERNAL TEST EQUIPMENT (ETE) SPECIFICATIONS OR SELECT ETE
C3
V3
V4
PDR SUPPORT
V7
A. SUPPORT PDR
DESIGN TEST EFFECTIVENESS ANALYSIS
F2
HARDWARE DESIGN
SOFTWARE DESIGN
TEST DATA
DOCUMENTATION
GFE DATA
FAILURE MODES
FREQUENCY OF OCCURRENCE OF EACH FAILURE MODE
A. PREDICT LEVELS OF FAULT DETECTION AND FAULT ISOLATION FOR THE SYSTEM, SUBSYSTEMS AND EACH UUT
F4
TEST REQUIREMENTS ANALYSIS
F1
CI AND UUT DESIGN DATA
DRAWINGS, FAILURE DATA, SPECIFICATIONS, THEORY OF OPERATION, MECHANICAL / ELECTRICAL INTERFACE, TESTABILITY ANALYSIS DATA
A. PRODUCE A VALIDATED TEST REQUIREMENTS DOCUMENT FOR EACH UUT
B. PRODUCE A DIAGNOSTIC SOFTWARE SPECIFICATION
C. TEST PROGRAM SET (TPS) DEVELOPMENT
TESTABILITY COSTS/ BENEFITS ANALYSIS
F3
WBS PER MIL-STD-881
CI AND UUT DRAWINGS
RELIABILITY DATA
A. DEVELOP LCC TESTABILITY BENEFITS AND COSTS

3

Full Scale Development Phase Testability Tasks



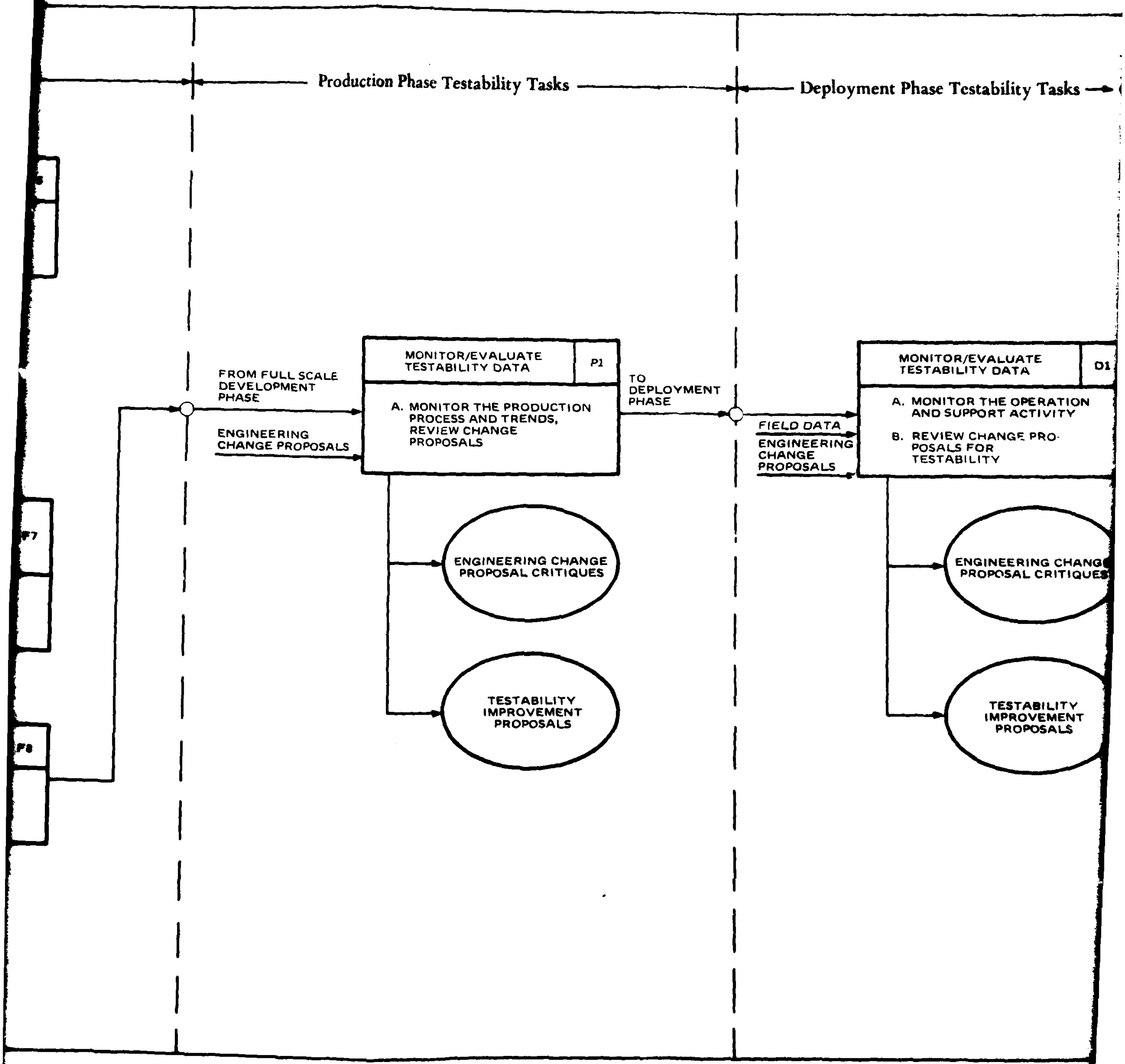


Figure 2-1. Testability Program Task Flow.

5

2.3.2 Testability Task Compendium Format and Characteristics

Each Testability Task Compendium consists of the following segments:

TASK REFERENCE NUMBER:
Identifies the task by number and traces directly to the Testability Flow Diagram.

PHASE:
Identifies the task's position in the procurement cycle. It traces directly to the Testability Flow Diagram.

FUNCTION:
Identifies the name of the activity block (the top line of each block in the flow diagram) which includes this task.

TASK TITLE:
Gives the name of the particular task. The title is also placed in the flow diagram with the task reference number.

TASK OBJECTIVE:
Provides a concise statement describing the goal(s) of the task.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:
Lists the design groups the testability engineer will coordinate within the task's performance, the input data he will need from the design groups and the output expected from him.

COST TRADEOFF INTER-RELATIONSHIPS:
Describes the equipment life cycle costs to be considered in performing and managing the task.

TASK SYNOPSIS:
Consists of three parts

(1) Task Requirements, which describe what has to be done to accomplish the task.
(2) Implementation, which describes the design techniques, tradeoff tools, optimization techniques and technical guidance needed to accomplish the task.
(3) Completion Criteria, which provides a check list, figure-of-merit or other criteria for use in judging the completion of the task.

For some tasks, certain detailed tools and techniques have been placed in appendices to the individual compendium.

Source material, where used, has in general been subjected to editing to fit the style and format of the task compendia. In most cases, the technical context and content of the source material has been preserved.

2.3.3 Application Notes

Three considerations should be noted in using the contents of the Testability Notebook.

(1) Most importantly, the system/unit designers themselves are to be encouraged to be testability engineering conscious. It is far more preferential that the design engineer be well acquainted with testability (and of course reliability, maintainability and supportability) design techniques as he is with performance design techniques, than for him to "answer" to or be driven by, a somewhat alien testability engineer. However, most designers need assistance, and they should be given the testability assist - ahead of their actual design effort - in terms of guidance for inclusion rather than requirements for correction of an already developed design. It is not proper to take the approach of achieving testability via ECP.

(2) There is a broad spectrum of possible program activity and task phasing, and that presented in the Testability Notebook should be taken as guidance only. Rearrangement of task phasings, inputs and outputs may well need to be accomplished for many programs which either have bypassed certain phases or for which testability engineering has been omitted from earlier phases.

(3) The full scope of all source data has been necessarily reduced to fit the Notebook. To have included full scope would have been self-defeating in terms of bulk and added confusion. The condensations in scope have been accomplished in three ways (additional to wording redundancy among sources); these are by editing into appendix format, by condensing for inclusion in the task synopsis and by noting for reference only. Notebook users are therefore encouraged to obtain copies of source data where added scope of treatment appears helpful.

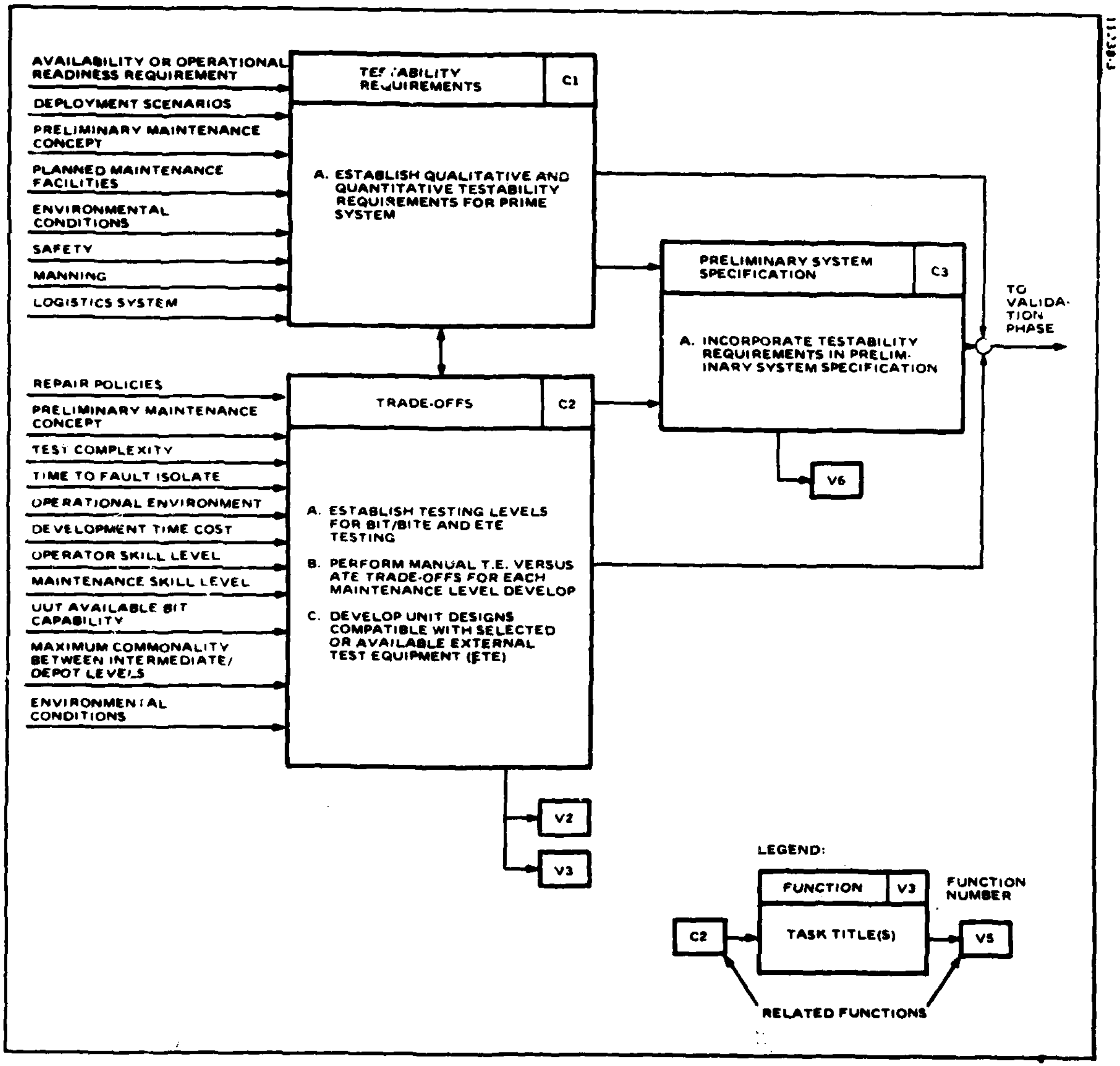
TESTABILITY TASK COMPENDIA

SECTION I - CONCEPTUAL PHASE

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES



Conceptual Phase Testability Tasks

TESTABILITY TASK COMPENDIUM
Task Reference Number C1A

PHASE: Conceptual

FUNCTION: Testability Requirements

TASK TITLE: Establish qualitative and quantitative testability requirements for prime system.

TASK OBJECTIVE: Establish testability requirements that are based upon operational requirements in sufficient time to guide the preliminary system design effort.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineer	● System Functional Concept	● Preliminary $\overline{T}$ Requirements
● Maintainability Engineer	● Preliminary Maintenance Concept, Maintenance Facilities	● Preliminary $\overline{T}$ Requirements
● Logistics Support	● Logistics System, Manning	● Preliminary $\overline{T}$ Requirements
● Safety Engineer	● Safety Considerations	● Preliminary $\overline{T}$ Requirements
● Program Manager	● Critique/approval of $\overline{T}$ Requirements	● Preliminary $\overline{T}$ Requirements

COST TRADEOFF INTER-RELATIONSHIPS: Timely development of the testability requirements defined herein will simplify subsequent tasks and is necessary to assure minimum program life cycle costs.

TASK SYNOPSIS:

1. Task Requirements.

1.1 Translate the operational readiness and/or equipment availability requirement into the following testability requirements:(1)

a. Maximum allowable time between the occurrence of a failure condition and the reporting of the failure (failure latency) for each mission function.

b. Degree of failure tolerance required for each mission function.

c. Maximum system downtime due to corrective maintenance actions at the organizational level.

d. Testing requirements of backup (standby) equipment and functions in order to accommodate system degraded mode requirements.

2. Task Implementation.

Establishing the testability requirements is an iterative process in which the testability requirements are optimized with respect to other system characteristics, e.g., BIT/ATE utilization, manual/automatic test equipment for system monitoring, and optimizing the mix of BITE, portable testers and maintenance shops to support organizational maintenance. The testability requirements established by this iterative process form the basis for the system specification testability requirements. Subsequently, functions C2 and C3 include tasks which treat (respectively) tradeoffs among the requirements/characteristics and the merging the testability requirements into the specification.

2.1 The qualitative and quantitative testability requirements should:

a. factor safety considerations into the requirements for failure detection and failure tolerance.

b. be based upon expected numbers and skills of operating and maintenance personnel.

c. be consistent with constraints imposed by the logistic system, including GFE support systems.

d. be consistent with the preliminary maintenance concept, deployment scenarios, environmental conditions and planned maintenance facilities.

2.2 The evaluation of operational considerations is the starting point for identifying system requirements.(2)

- Review operational concept-threat, mission analysis, operating environment.
- Determine prime function of the system and relate it to the operational concept.

- Determine operational limiting constraints. These parameters establish the maximum repair time that enable all operational scenarios to be accomplished.

  - Availability
  - Critical failure mode
  - Operational fault detection/fault isolation (FD/FI)
  - Failure latency time
  - Isolation level
  - Allowable downtime

2.3 Many factors affect testability and should be kept in mind during the tradeoff process throughout the program. (3) The important factors relating to operation and maintenance scenarios are:

- Mission Success

  - Reliability
  - Confidence Level (Failure Detection Probability, Minimum False Alert)

- Availability

  - Readiness Test Time
  - Maintainability

    - MTTR

      - Required Skill Level (Training Time and Cost)
      - Complexity (Modularization, Standardization)
      - Fault Isolation Time (ATE Software and Machine Loading Time, BIT Costs; Manpower)
      - Fault Correction Time
      - Verification Time (Operator Time, BIT and ATE Software/ Hardware Costs)
      - Test Recycling Time Due to Non-Detectable Faults (Manpower, ATE Machine Loading Time)

- Logistics

  - Weight, Size, Shape
  - Mission Duration
  - Distance from Supply Source
  - Storage Limitations
  - Software, Manuals, Tools Required
  - Manpower Requirements
  - Test Instrumentation Maintenance, Tools
  - Test Recycling Time Due to Non-Detectable Faults (Manpower, Test Instrumentation)

- o Expansion Capability of System (BIT, ATE)
- o Life Cycle Costs
  - Hardware
  - Software
  - Maintenance
  - Administrative

3. Completion Criteria.

3.1 The task is completed when there is agreement between testability engineering, the interfacing groups and the program manager that the iterative process has established the optimum prime system testability requirements.

TESTABILITY TASK COMPENDIUM

Task Reference Number C2A

PHASE: Conceptual

FUNCTION: Tradeoffs

TASK TITLE: Establish testing levels for BIT/BITE and ETE testing

TASK OBJECTIVE: Identify the boundary or level to which BIT/BITE testing will be utilized.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering	● System Level BIT/BITE/ETE Approaches	● $\overline{T}$ Guidelines
● Design Engineering	● Hardware (HW) BIT/BITE Alternatives	● $\overline{T}$ Guidelines
● Application Software	● Software (SW) Alternatives	● $\overline{T}$ Guidelines
● Life Cycle Costing	● Tradeoff Costing Assistance	● Costing Data and Parameters
● Maintainability Engineering	● Proposed Maintainability Program Plan and Integrated Logistics Support Plan	● $\overline{T}$ Guidelines

COST TRADEOFF INTER-RELATIONSHIPS: The recurring and nonrecurring costs associated with various test mixes need evaluation in this task. Generally, software approaches (BIT and ATE) exhibit competitive acquisition costs but lower maintenance costs (including ECP impacts) compared to approaches which rely more heavily on unique test systems.

TASK SYNOPSIS:

1. Task Requirements.

The task is performed to identify the boundary or level to which BIT/BITE testing will be utilized.

1.1 BIT/BITE/ETE utilization.(1) The norm is to establish a two-level, preferably automatic, test concept which is driven by the natural break between BIT capabilities and ETE capabilities.

a. Built-in test should be utilized to provide initial fault detection for a system or equipment and to provide initial fault isolation to a small group of modules.

b. Off-line or External Test Equipment (preferably approved ATE) should be used to provide fault detection for a module as a Unit Under Test (UUT) and to provide fault isolation to components within a module.

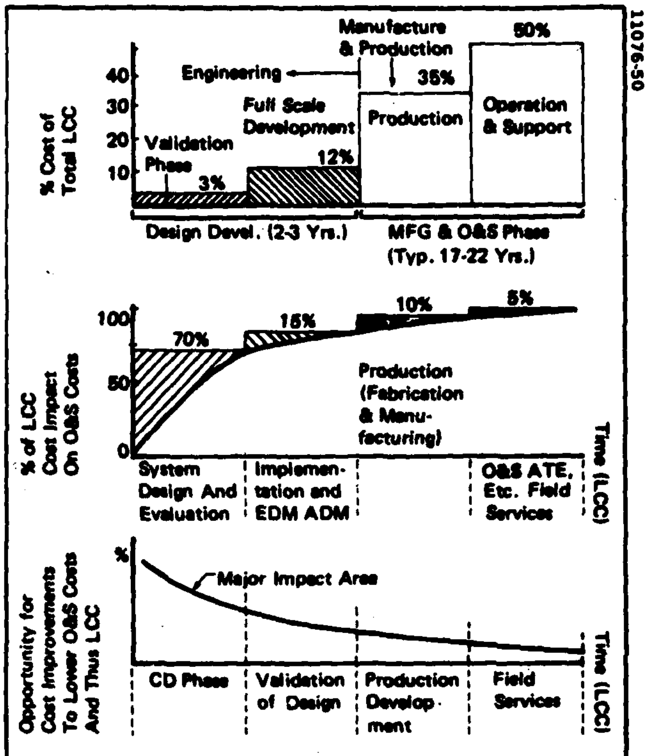
1.2 BIT/BITE/ETE Coordination.(1) Strive for maximum utilization of the system BIT capability which is distributed to individual UUTs in determining off-line test requirements for the UUTs.

1.3 Timeliness Requirement. System BIT/BITE/ETE definition must be accomplished early in the conceptual phase. The following paraphrased excerpt illustrates the cost effectiveness of early and proper decisions: (4)

> "The total cost of a sample weapons system over its life cycle-approximately 20 years or 100,000 hours - has an interesting cost parameter distribution. The engineering validation phase requires only 3 to 4 percent of total life-cycle cost, while the preproduction phase takes 12 percent - a total of just 15 percent for the entire prototyping/preproduction period. Production costs typically average 35 percent; operation/support costs comprise 50 percent (Fig 1). The 17-22 year ownership of the system, not the initial acquisition expense, represents the primary cost factor.
>
> "Further DoD studies demonstrate that once design is complete, the best ETE approach can reduce life cycle cost by at most 5 percent; production testing can achieve a 10 percent cost enhancement, and hardware design at most 15 percent. Proper system design and architecture with sufficient provision for built-in testability may influence upwards of 70 percent of the life cycle cost."

1.4 Maintenance Time and Availability Impacts.(5)

Early identification of prime system characteristics and test subsystem characteristics is essential for the test subsystem to be effective in performance monitoring, fault detection, and fault isolation. Figure 2 illustrates these key points.



Intensive total system design analyses are essential for lowest O&S cost, by proper design for PM, FD/FL (BIT/BITE) integral with system (70% impact potential).

Operation and support (O&S) costs of a weapon system have major impact on total life cycle cost (LCC) (50%).

Figure 1. Major Life Cycle Cost Drivers

- Mean corrective maintenance time ($\overline{M}_{ct}$) is a major determinant in selecting candidate test subsystems.

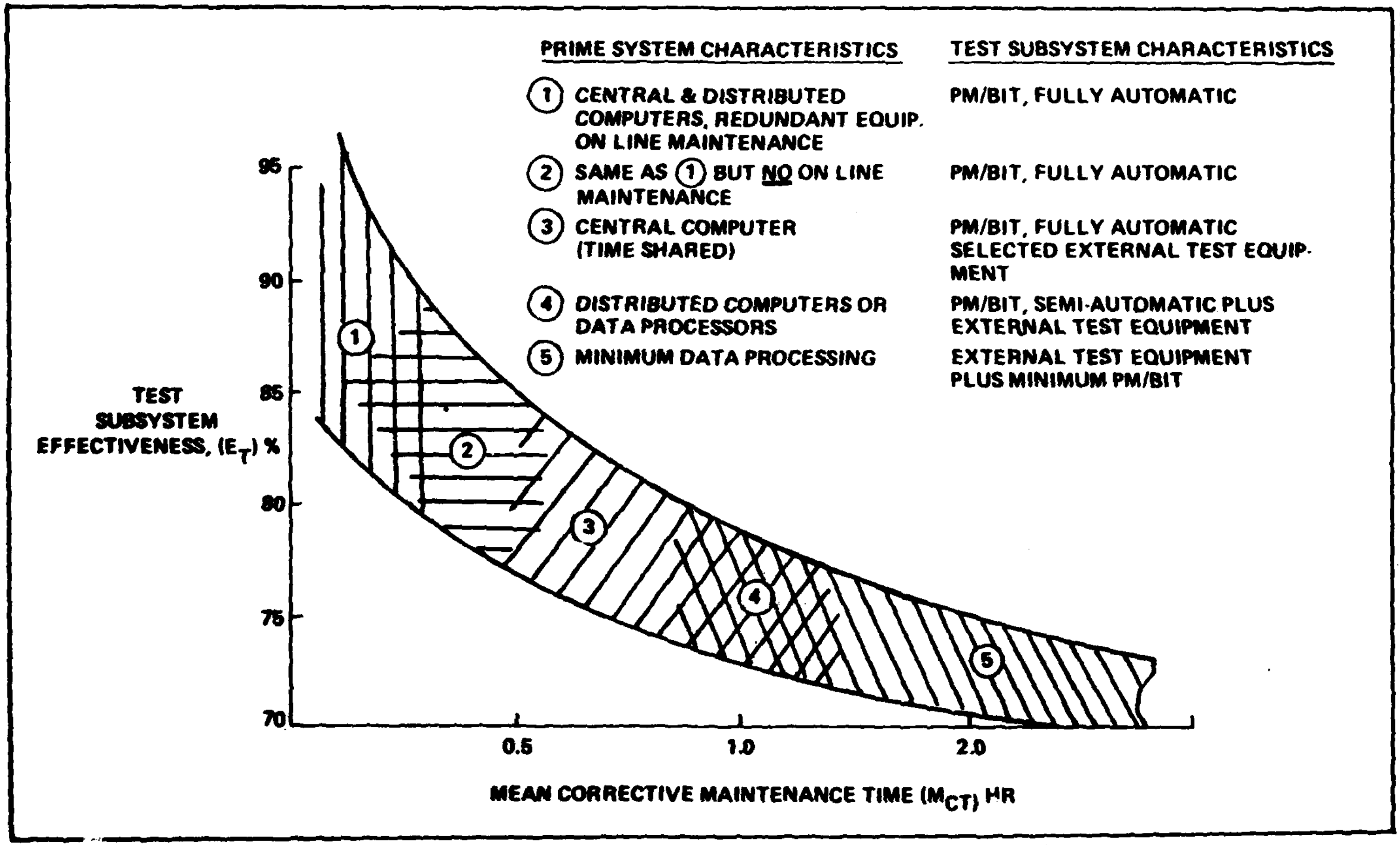
- The prime system's characteristics are a key determinant in selecting the optimum performance monitoring (PM) and BIT effectiveness goal.

2. Implementation.

2.1 BIT/BITE Selection Factors(6)

a. Prime equipment can be tested either with BIT or with external test equipment. For predominantly digital systems, the hardware BIT concept is the more cost effective approach. Factors reducing the cost of BIT logic include:

11076-4



PRIME SYSTEM CHARACTERISTICS	TEST SUBSYSTEM CHARACTERISTICS
① CENTRAL & DISTRIBUTED COMPUTERS, REDUNDANT EQUIP. ON LINE MAINTENANCE	PM/BIT, FULLY AUTOMATIC
② SAME AS ① BUT NO ON LINE MAINTENANCE	PM/BIT, FULLY AUTOMATIC
③ CENTRAL COMPUTER (TIME SHARED)	PM/BIT, FULLY AUTOMATIC SELECTED EXTERNAL TEST EQUIPMENT
④ DISTRIBUTED COMPUTERS OR DATA PROCESSORS	PM/BIT, SEMI-AUTOMATIC PLUS EXTERNAL TEST EQUIPMENT
⑤ MINIMUM DATA PROCESSING	EXTERNAL TEST EQUIPMENT PLUS MINIMUM PM/BIT

Figure 2. State-of-the-Art (Circa '75-'80) Prime System vs Test Subsystem Characteristics

(1) Efficient coding techniques minimize the added logic complexity for BIT.

(2) Use may be made of unfilled PC board package positions for at least a portion of the BIT logic, so that the cost penalty for BIT is considerably less than the increased logic complexity.

(3) Added design effort is minimal since test constraints necessarily will be imposed and the built-in test design is accomplished with systematic logic design procedures.

b. The BIT has the objectives of 1) detecting subsystem faults, 2) isolating to one subsystem module and 3) providing aid in isolation to a specific faulty IC component.

c. Using BIT hardware instead of software, a test method can be developed to fault detect and fault isolate to a digital subsystem and to the faulty module therein. The test procedure compares module output patterns (obtained by stimulating each module with fixed input patterns) to the known responses of fault-free modules. This test technique is commonly used in programmed test equipment for static testing of digital modules. Generally, higher level assemblies (entire subsystems) are too complex for this pattern comparison technique and, therefore, require complex test sequences and/or parametric testing. If each subsystem is self-testing and self-isolating to the module level, external test equipment and even certain levels of maintenance can be greatly simplified or eliminated. External test equipment requirements may be simplified to provide only power, cooling, initialization, and, optionally limited display or comparison of module test results. Further, the self test may be performed in an operational system, isolating operational faults not detected in maintenance. Only milliseconds are required for BIT performance.

2.2 ETE Selection Factors. (7)

There are two main functions which must be performed in ETE concept formulation. These are establishing performance monitoring need and determining the degree of off-line ETE at organizational, intermediate, and depot levels.

a. There are three subsets of the Performance Monitoring Needs to be considered:

- The level to which system and platform performance monitoring will be performed.
- The environmental factors (e.g., safety, damage control, EMCON-Electromagnetic Emission Control) to be monitored.

- The system configurations which need to be displayed e.g., communication channels, electrical power plant status).

From the information available at this stage of development, it is possible to start formulating the performance monitoring needs of the system. Such things as BIT, BITE, ON-Line Test, and Self Test needs can be formulated. These needs can then be blended into the definition of the prime system and the support system. This effort will produce a preliminary integrated prime hardware and support system design. Subsequent iterations with the other factors/ operators in the definition process may change the hardware definition but certain general attributes remain constant. It is quite feasible to define BIT/BITE or On-Line Test during the first cut at a hardware definition (design) and have it remain relatively stable through the development cycle. Because of the highly integrated nature of BIT/BITE and the prime hardware it is almost mandatory that it be specified/defined during the conceptual phase of development. It is too late to wait until the full scale development or production phases to introduce BIT/BITE or On-Line ETE. Performance monitoring needs should also be identified early in the development cycle so that the requisite sensors and monitoring points can be designed/built into the prime hardware. The decision to use full or partial off-line ETE can and should also be made this early in the development cycle, to allow an evolutionary selection process.

b. In addressing the degree of off-line ETE for each maintenance level, a variety of iterations and the tradeoffs are possible with impact on the full range of ETE concepts. This tradeoff of alternative concepts is a factor in the definition of ETE alternatives. Following are the steps taken in the definition of alternatives.

(1) Propose Generic ETE Types. Generic ETE types are proposed which are compatible with each maintenance concept under study as a first step in the selection process. Possible generic ETE options for a system can be made up from one or a combination of the following:

- Built-In Test (BIT)
- Built-In Test Equipment (BITE)
- Other on-line test systems
- Off-line test systems

(2) Identify On-Line Test Requirements. Prime system and mission needs will have to be analyzed to determine the need for on-line monitoring. In general, from an operational and maintenance viewpoint, on-line monitoring (or testing) is the most desirable mode of operation. The tradeoffs involved are cost, technical impact on the design, and the operational requirements for the mission.

(3) Identify The Degree of Off-line Test Requirements. Many of the selection considerations for on-line test requirements are also applicable to off-line testing. The degree of off-line testing will largely depend on decisions regarding specific maintenance levels and locations (i.e., organizational, intermediate and depot).

At the conceptual level of development the ETE alternatives can only be matched with the degree of detail available on the prime hardware, its cost estimate, and the technical definition of detail of the hardware alternatives. It is quite likely that several possible acceptable alternative ETE concepts will still exist after the end of the concept phase.

2.3 Documentation.

The aim of this task is to provide the most efficient use of and the most cost effective mix of BIT/BITE and ETE over the program life cycle. A plan for the level of BIT within the system design and the ETE required should be developed. The plan may contain more than one alternative that meets the operational requirements. Selection of the optimum mix (alternative) will come about as the design firms up in later phases.

3. Completion Criteria.

The task is completed at government approval of the documented plan for achievement of the optimum BIT/BITE/ETE mix.

TESTABILITY TASK COMPENDIUM
Task Reference Number C2B

PHASE: Conceptual

FUNCTION: Tradeoffs

TASK TITLE: Perform manual test equipment vs ATE tradeoffs for each maintenance level.

TASK OBJECTIVE: Define the mix of Manual and Automatic Test Equipment at each maintenance level.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
• Design Engineering	• System/Equipment Requirements, BIT Capability Data	• HW Testability Requirements
• Test Engineering	• Test Complexity Factors	• Testability Requirements
• Test Equipment Engineering	• Manual Test Equipment and ATE Capabilities, Requirements, & Costs Data	• Test Equipment Testability Requirements
• Application Software	• SW Program Requirements	• SW Testability Requirements
• Maintainability Engineering	• Maintainability Program Concept, Integrated Logistics Support Concept	• Testability Requirements
• Life Cycle Costing	• Costing Guidelines and Cost Analysis Results	• Cost parameters

COST TRADEOFF INTER-RELATIONSHIPS: Non-recurring and recurring costs for acquisition, use and logistic support of test equipment and test programs are affected by this decision. Benefits and penalties occurring in all phases from Validation through Operations and Support must be considered.

TASK SYNOPSIS:

1. *Task Requirements.*

Develop an integrated test policy for the system, trading use of manual versus use of automatic test equipment (ATE) for each maintenance level. Consider test complexity, repair policy, fault isolation time, functional test time, operational environment, logistic support requirements, development time, skill levels, and all acquisition and ownership costs.

2. Implementation.

2.1 Test Tradeoffs.(1) Decisions regarding the type of test equipment to be used for system monitoring and maintenance should be made based upon repair policies and overall maintenance plans. Tradeoffs should be made for test requirements at each maintenance level, considering test complexity, time to fault isolate, operational environment, logistic support requirements, development time and cost. The degree of testing automation should be consistent with the planned skill levels of the equipment operators and maintenance personnel.

2.2 Considerations. Include the following considerations in the tradeoffs, where applicable:

a. Costs to buy or develop test equipment (T.E.)
b. Skill level of personnel required to support T.E.
c. Development time for selected T.E.
d. Adaptability of T.E. to design changes
e. Manning requirements of T.E.
f. Utilization of T.E.
g. Programming requirements and costs of T.E.
h. UUT fault isolation and repair time using T.E.
i. T.E. failure rates, fault isolation requirements and time, and repair time
j. Total LCC of selected T.E.
k. Ability of T.E. to meet system test requirements
l. Prime system availability and maintainability requirements
m. Special tester and/or interface requirements for system UUT's such as micro-processor and hybrid boards
n. Test equipment failure rates (availability)
o. Other contractually specified requirements

2.3 Test Equipment Costs. Tradeoffs should evaluate the proposed test methodologies for total LCC. This evaluation should include initial price (hardware, software, adapters, and patchboards), programming time, future test requirements, and system throughput. The form shown in Table 1(8) provides a form of basic guidance which is adaptable for use in cost comparisons of various systems.

Either an existing or a candidate new system may be set up as a baseline for comparison purposes. Acquisition, use and support costs should be segregated for separate viewing, so that choices may be between competing systems on the basis of equivalent life cycle cost elements, individual and total.

TABLE 1. CALCULATION OF COST OF OWNERSHIP	Baseline Test Equipment System	Alternative A Test Equipment System	Alternative B Test Equipment System	
Section I - System acquisition				
(a) Hardware costs				
(b) Software costs				
(c) Support costs				
Total System Acquisition Costs				
Section II - "Production" testing				
(a) Adapters, patchboards, fixtures, etc.				
(b) Test program cost				
(c) Set-up cost				
(d) Test cost				
(e) Fault isolation cost (troubleshoot)				
(f) Set-up retest cost				
(g) Retest cost				
Total production test cost				
Section III - Data analysis and reports				
(a) Quality control reports				
(b) Logistics and field service reports				
(c) Configuration management reports				
Total data analysis and report costs				
Section IV - Other ownership costs				
(a) Refurbishment cost				
(b) Multistation capability				
(c) Remote station control				
Total other ownership costs				
TOTAL COST OF OWNERSHIP				

2.4 Achievement Definitions. Successful completion of this task will result in selection of the most cost effective (in terms of total LCC) test system. In the conceptual phase, the selection process may not achieve selection of a singular system but should at least define a peculiar type of test system, leaving the final selection for validation phase. The type of system chosen should be documented, along with the rationale used to arrive at the chosen system for evaluation in the validation phase.

3. Completion Criteria.

The task is completed upon Government concurrence in the documented selection of test system.

TESTABILITY TASK COMPENDIUM

Task Reference Number C2C

PHASE: Conceptual

FUNCTION: Tradeoffs

TASK TITLE: Develop unit design features compatible with selected or available ETE.

TASK OBJECTIVE: Control prime equipment/ETE interface compatibility and associated costs.

DESIGN DISCIPLINE TESTABILITY ($\bar{T}$) INTER RELATIONSHIPS:

$\bar{T}$ INTERFACE	INPUT TO $\bar{T}$	OUTPUT FROM $\bar{T}$
● System Engineering	● Performance Data	
● Design Engineering	● UUT Interface Data	● ID Related Requirements,* and Mechanizations
● Test Equipment Engineering	● ETE Interface Data	● Compatibility Mechanizations
● Life Cycle Costing	● LCC Data and Requirements	● Data Supporting LCC

* Ideally, the interface should be established between the design and test equipment engineering functions, with monitoring only by system and testability engineering.

COST TRADEOFF INTER-RELATIONSHIPS: The control of the unit/test equipment interface requires some added non-recurring effort in design and engineering. Reduced efforts as a result occur downstream in both non-recurring and recurring costs of interface devices, adapters, procedures, technical data, training and manpower expended in maintenance test and repair. The cost breakeven occurs, and savings begin to accrue, during validation or full scale/engineering development phases.

TASK SYNOPSIS:

1. Task Requirements.

1.1 The requirement for compatibility control applies to all external test equipment, whether manual or automatic, general or special. The concept of control is applied to minimize the complexity and cost of maintenance testing, without degrading the operational performance of the prime system. In this synopsis, the term ETE can be construed to include automatic, manual, general or special purpose test equipment external to the BIT/BITE within the prime system. Controls are established in the conceptual phase to effect:

a. Control of the design of prime equipment interfaces to meet existing ETE interface specifications.

b. Control of the design of new ETE to meet prime equipment interface specifications.

2. Implementation.

Published compatibility techniques deal primarily with automatic test equipment, but are generally readily applicable to other types of test equipment.

2.1 Fundamental Considerations. Policy imposed by the acquisition agency or program manager may set the mode of the task by designating either that existing test equipment be utilized or that new test equipment should be designed. In any event, some combination of three overall methodologies will probably be applicable to any particular project:(1)

a. Control of the design of prime equipment interfaces to meet existing ETE interface specifications.

b. Control of the design of new ETE to meet prime equipment interface specifications.

c. Design of interface devices to bridge incompatible prime equipment and ETE.

(As a first priority, it is usually most effective to utilize ETE systems existing in inventory or already under development which will meet support requirements.)

2.2 The following is a checklist that is applicable to compatibility tasks. The checklist is stated in the form of questions relating to commonly recognized ETE compatibility design parameters.(9)

a. ETE COMPATIBILITY CHECKLIST

(1) System Modularity. Is the system functionally modularized to the UUT level of assembly/disassembly?

(2) Functional Independence. Are the system and its UUT's capable of being tested without stimulation by another system or UUT and without simulation of another system or UUT?

(3) Adjustments. Are system/UUT adjustments (e.g., trimming, tuning, alignment) required while testing on ETE? (An adjustment includes any action that changes variable components such as potentiometers, variable capacitors, inductors, transformers, etc., that affect operation of the equipment.)

(4) Ancillary External Test Equipment. Is additional external equipment required to generate a stimulus or to monitor response signals?

(5) Environmental. Will the system or the UUT's require special environmental considerations during test on the ETE, such as vacuum chambers, oil baths, shake tables, ovens, cooling air and screen rooms?

(6) Stimulus and Measurement Accuracies. Are the stimulus and measurement accuracies required for high confidence test available in the candidate ETE?

(7) Test Point Adequacy. Are sufficient test points provided for non-ambiguous fault isolation and for monitoring redundant circuits and BIT circuits?

(8) Test Point Characteristics. Are test point impedance and voltage levels compatible with the ETE interface?

(9) Test Point Isolation. Will damage to a UUT result from a short circuit between any test point and ground? Will wideband noise impressed on the test point degrade performance?

(10) Power and Load Requirements. Are the current and voltage required for system/UUT power and the loads required to absorb the output power available at the ETE?

(11) Warm-Up. Will the system and/or any UUT require warm-up on ETE to ensure accurate test?

(12) Access. Is internal access adequate for visual inspection and manipulative actions?

(13) Packaging. Is access to UUT adequate? Can the UUT be removed and replaced easily? Are special tools required?

(14) Safety (Personnel). Will the maintenance action require personnel to work under hazardous conditions such as close proximity to high voltage, radiation, moving parts, or high-temperature components, etc.

(15) Connector Standardization. Are standard connectors used on the system/UUT?

(16) Connector Keying and Accessibility. Are the system/UUT connectors keyed to preclude inserting any connector into the wrong receptacle, and are they readily accessible for quick connection and removal?

(17) EMI or RFI. Will the system require special testing due to EMI or RFI performance characteristics?

2.3 Design Objective:

Successful completion of this task results in a proposed UUT/ETE interface design that is compatible with both the prime system and the ETE. Interface devices (adapters) should be held to a minimum in quantity and complexity.

3. Completion Criteria.

Documentation should be developed describing the test interface with justification for the designs and test equipments chosen. The task is complete when the government acquisition manager accepts the design factors and candidate equipment.

TESTABILITY TASK COMPENDIUM
Task Reference Number C3A

PHASE: Conceptual

FUNCTION: Preliminary System Specification

TASK TITLE: Incorporate testability requirements in preliminary system specification

TASK OBJECTIVE: Refine the requirements developed in Task Number C1A into specifiable goals for test and testability requirements and include the results in the preliminary system specification.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering	● System Requirements and Tradeoffs	● Testability Requirements

---

COST TRADEOFF INTER-RELATIONSHIPS: There are no significant cost impacts in the specification input task. Over-specification should be avoided because of obvious cost ramifications, but the depth of testability specified must be adequate to avoid subsequent costly re-design because of field test and maintenance complexities.

TASK SYNOPSIS:

1. Task Requirements.

1.1 The testability goals/requirements should include, but not be limited to, the following subjects:(1)

a. Requirement for status monitoring.

b. Definition of the failure modes specified to be the basis for test design.

c. Requirement for failure detection (failure coverage, failure latency) using full test resources.

d. Requirement for failure detection using built-in test resources.

e. Requirement for failure detection using only passive monitoring.

f. Requirement for limiting false alarm rate.

g. Requirement for failure localization to a subsystem/equipment using built-in test

h. Requirement for failure isolation to one or more number of modules using built-in test. The requirement may be expressed in terms of percentage of modules in a subsystem/equipment.

i. Requirement for failure localization/isolation times.

j. Restrictions on built-in test resources in terms of hardware size, weight and power, memory size and test time.

k. Requirement for BIT hardware reliability.

2. Implementation.

Functions C1 and C2, if performed, will have provided baseline data for use in the specification input_task. If there has been no prior interface, development of the T requirements for inclusion in the system specifications may need to be started on a qualitative level, then, given the development of interfaces with system design personnel, developed to a quantitative level.

2.1 System Specification Flow Diagram.(2)

The process for development of the preliminary system specification may be regarded as a flow of tasks as depicted in Figure 1. This flow process is comprised of a series of subtasks that upon completion lead to completion of the preliminary system specification. Certain testability features such as observability and controllability are not specified per se but are built into the system specifications through the system test requirements, MTTR requirements, and in the fault detection and fault isolation (FD/FI) requirements. The flow diagram in Figure 1 recapitulates all the testability functions and tasks of the conceptual phase.

2.2 Test Subsystem Performance Specification, Basic Testability Contents (5)

To optimize the subsystems performance and design, the following prime system and test subsystem performance parameters must receive due consideration in developing the system specification:

a. Operational demand and criticality

b. Mission duration and operational modes

c. Mission reliability

d. BIT MTBF

e. Maximum turn-around time (as applicable)

f. Allowable mean down time (as applicable)

g. Expected mean logistics and administrative delay times ($\bar{M}_l$ and $\bar{M}_a$)

h. Mean corrective maintenance time ($\bar{M}_{ct}$)

i. Test subsystem effectiveness (E_T)

j. Percentage of false alarms, no defect removals (FA)

2.3 Detail Testability Requirements. (10)

The following list of 17 generic groupings are the BIT/ETE Figures-of-Merit found in a survey of successful system specifications.

a. fraction of faults detected:

(1) percent of all faults automatically detected by BIT/ETE
(2) percent of all faults detectable by BIT/ETE
(3) percent of all faults detectable on-line by BIT/ETE
(4) percent of all faults and out-of-tolerance conditions detectable by BIT/ETE
(5) percent of all faults detectable by any means

b. fraction of false alarms

(1) rate at which false indications occur (per 10^6 hours)
(2) percent of indicated failures caused by actual failures
(3) percent of BIT/ETE indicated failures caused by actual failures
(4) percent of BIT/ETE fault isolations to the wrong UUT

c. fraction of false status indications

(1) percent of erroneous BIT indications

d. mean fault detection time

(1) time to indicate a fault once it has occurred
(2) time to detect a fault once it has occurred

e. mean BIT/ETE running time

(1) time to verify that a failure has occurred/or has been repaired using BIT/ETE

f. frequency of BIT/ETE executions

(1) percent of all equipment functions tested

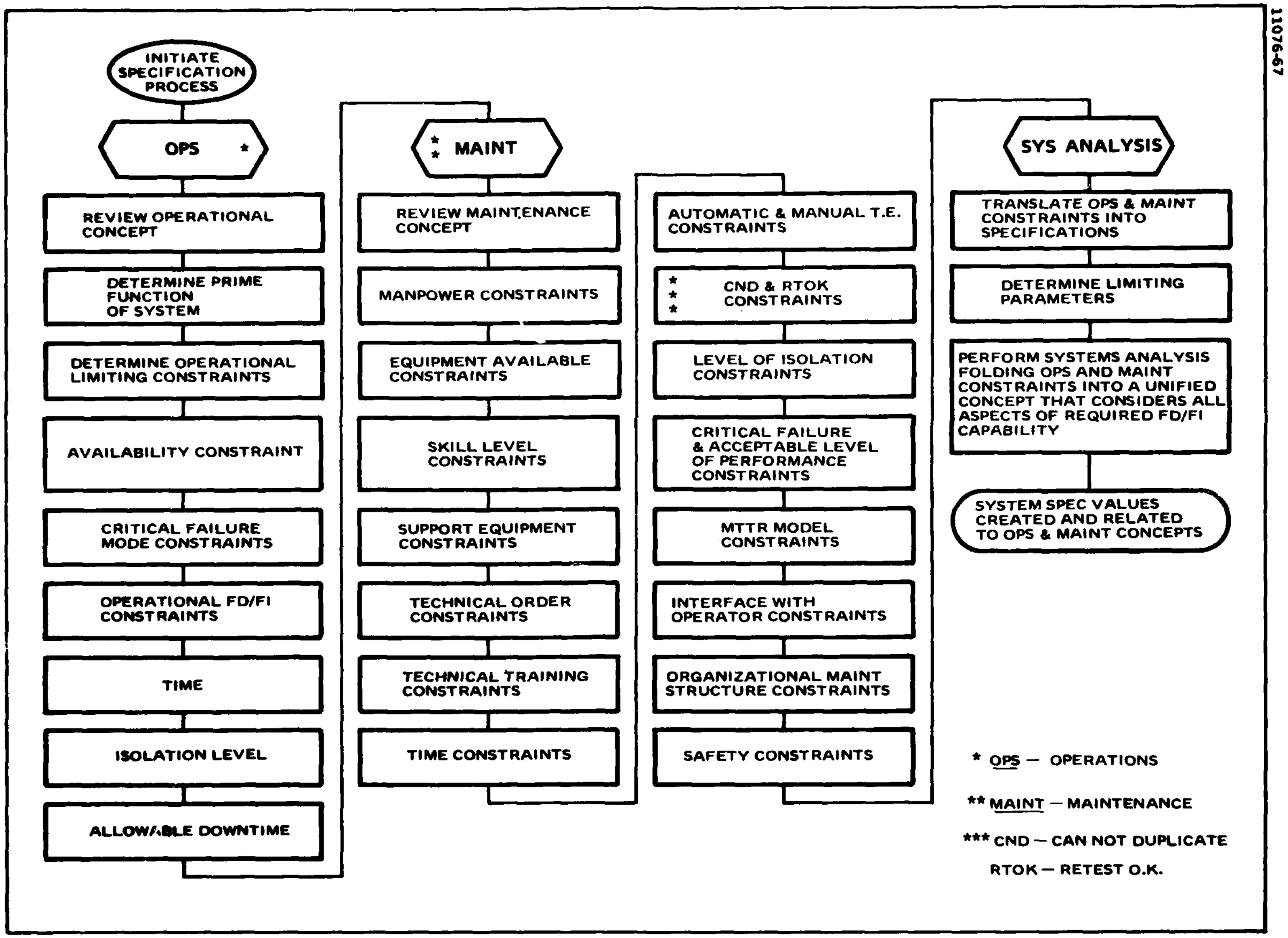


Figure 1. Specification Process (27)

g. test thoroughness

(1) percent of all equipment functions tested

h. fault isolation resolution

(1) isolation of P_1 percent of the failures to X_1 UUT's, P_2 percent of the failures to X_2 UUT's and so on, with any fault isolation method

(2) isolation of all faults to less than or equal to some maximum number of UUT's

(3) isolation of P_1 percent of the failures to X_1 UUT's, P_2 percent of the failures to X_2 UUT's, and so on, with BIT/ETE

(4) isolation of a specified percent of the failures to less than or equal to a specified quantity of UUT's at the various maintenance levels

(5) isolation of a specified percent of the failures down to less than or equal to a maximum number of plug-in modules

(6) isolation semi-automatically to a certain percent of all faults down to a specified number of UUT's

i. fraction of faults isolated

(1) isolate a specified percent of failures that occur within a specified maximum time

(2) isolate a failure down to a replaceable level, within a specified average time

(3) isolate a failure down to a replaceable level within a specified time once the fault isolation process has been initiated

k. maintenance personnel skill level

(1) all maintenance actions must be capable of being performed by a specified quantity of maintenance personnel with a specified skill level, at various maintenance levels

(2) BIT/ETE must be designed for use by a specified minimum skill level technician

l. BIT/ETE mean-time-to-repair

   (1) mean-time-to-repair ETE
   (2) mean-time-to-repair monitoring/fault isolation functions

m. BIT/ETE mean time between failures

   (1) mean time between failures of monitoring/fault isolation functions
   (2) mean time between failures of ETE only

n. BIT/ETE availability

   (1) monitoring/fault isolation functions should be operating with a specified probability of survival

o. mean-time-to-repair

   (1) system/equipment MTTR and maximum repair time
   (2) system/equipment MTTR and maximum repair time at various maintenance levels

p. availability

   (1) inherent availability
   (2) operational availability

q. active memory allocated for BIT/ETE functions

   (1) monitoring/fault isolation functions shall take up a specified percent of active computer memory

2.4 Reference Examples.

a. Reference (11) contains a sample on-board test system specification for part of the B1 aircraft's central integrated test system (CITS). CITS provides on-aircraft information relative to the health of sub-systems by in-flight monitoring and providing stimulus for ground testing.

b. Reference (12) contains sample testability paragraphs for system specification.

3. Completion Criteria.

The task is successfully completed by incorporating the testability goals/requirements typified by Paragraph 1, Task Requirements, and adapting those applicable portions of Paragraph 2, Implementation, into the preliminary system specification. The task is therefore judged complete on government approval of the preliminary system specification.

CONCEPTUAL PHASE BIBLIOGRAPHICAL REFERENCES

1. A Study of Testability Standardization for Electronic Systems and Equipment prepared by Testing and Technology Office, NSWC and Command & Control Applications Branch, NOSC for NESC, code 304, pg. 6, 7, 8

2. Diagnostic Specification - A Proposed Approach by Lt/Col. William H. Carroll, Jr., Maj. Vincent L. Linden, and 2Lt. Clarence R. Waldo; AFTEC/LG, Kirtland AFB, NM 87117, pg 6

3. Built-in-Test (BIT) Design Guide, NAVMATINST 3960.9A dtd 19 September 1979, pg. 3-29

4. Built-in-Test and VHSIC/VLSI Technology by J.M.H. Heines; Electronic Test, October 1980, pg. 60

5. Design Guidelines and Optimization Procedures for Test Subsystem Design RADC-TR-80-111, pg. 33

6. An Advanced Fault Isolation System for Digital Logic by N. Benowitz, D.F. Calhoun, G.E. Alderson, J.E. Bauer and C.T. Joeckel; IEEE Transactions on Computers, Vol. C-24, No. 5, May 1975 and Fault Detection/Isolation Results from AAFIS Hardware Built-in-Test by N. Benowitz, D.F. Calhoun, and G.W.K. Lee; NAECON '76 Record, pg. 220.

7. Procedure for the Selection of Automatic Test Equipment (ATE) dtd August 1974; ATEMAT(MATO3T), Headquarters Naval Material Command, pg. 9

8. To Test Hybrid PC Boards by P. Jackson; Electronic Design 24, November 22, 1975, pg. 121.

9. MIL-STD-2076 dtd 1 March 1978; Unit Under Test Compatibility with Automatic Test Equipment, General Requirements for, Appendix B.

10. BIT/External Test Figures of Merit and Demonstration Techniques; RADC-TR-79-309, pg. 8

11. B-1 Specification Format Basic Procurement Specification dtd 24 June 77; Los Angeles Division, North American Rockwell Corporation, pg. 12-16, 24, 63

12. A Framework for Designing Testability into Electronic Systems by W.L. Keiner; TR3826, Naval Surface Weapons Center, Dahlgren Laboratory (Code K43), Dahlgren, Virginia 22448, pg. 37

TESTABILITY TASK COMPENDIA

SECTION II - VALIDATION PHASE

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

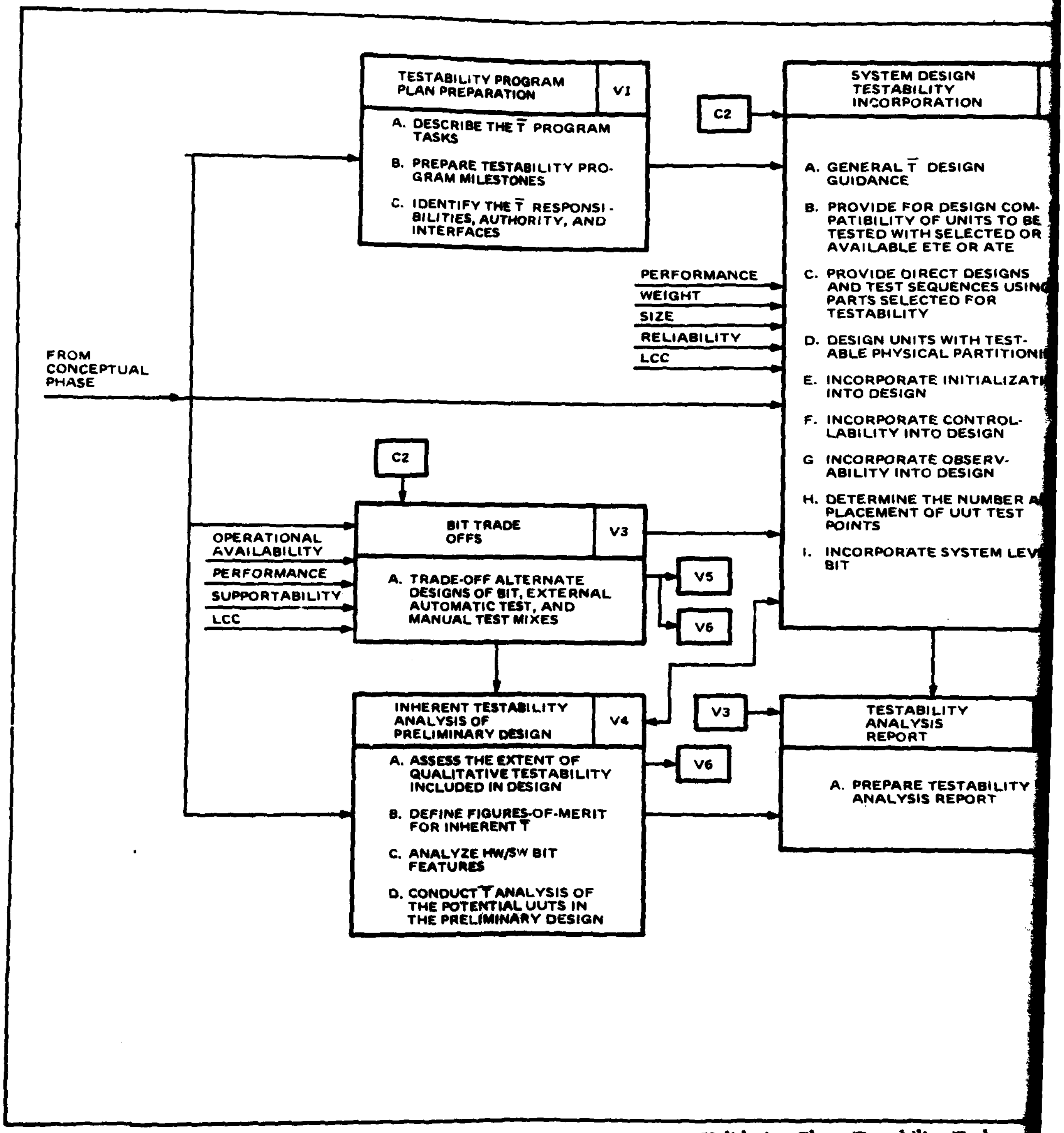
LIST OF ILLUSTRATIONS

TABLE OF CONTENTS (Continued)

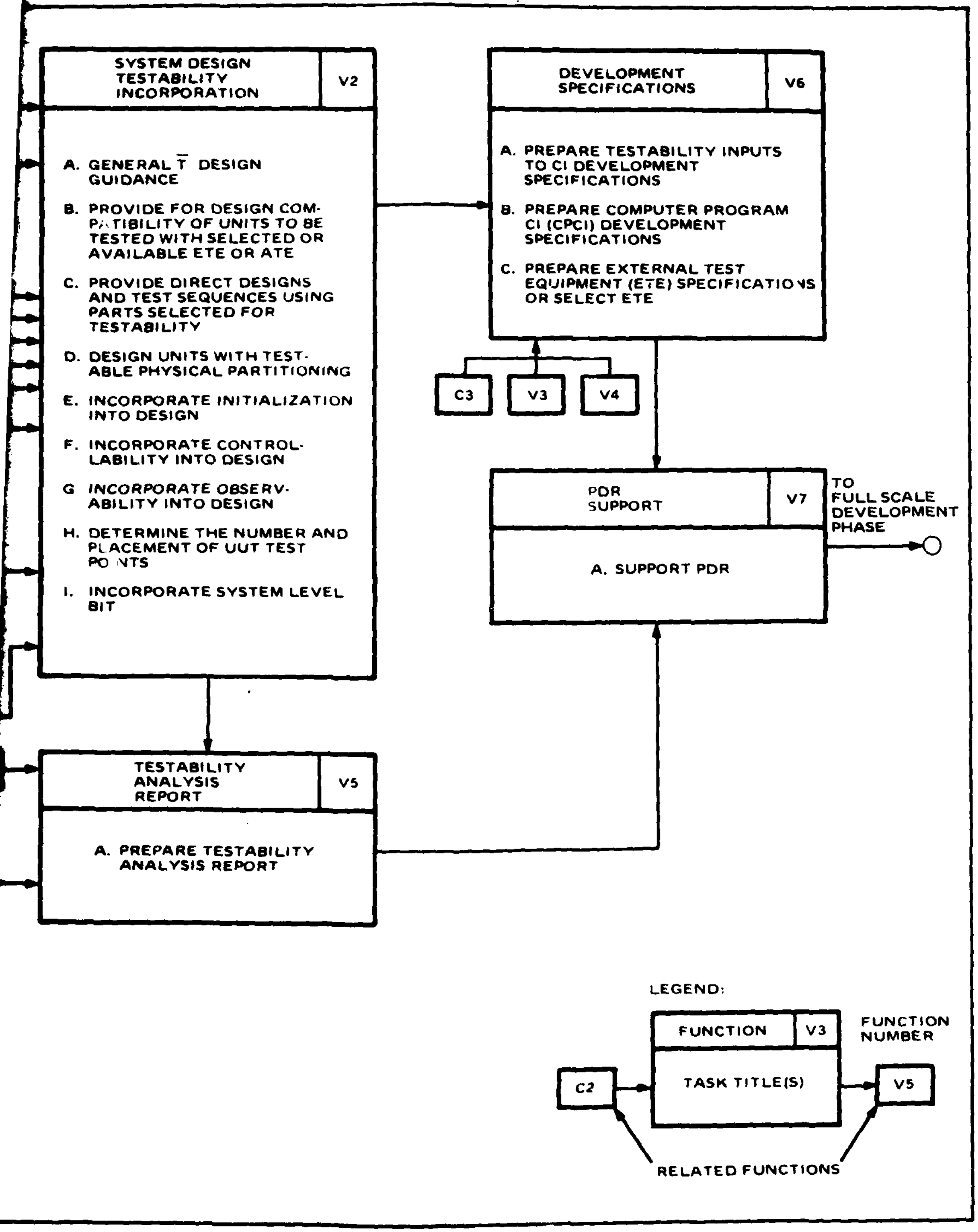
LIST OF ILLUSTRATIONS

LIST OF TABLES

*Compendia Appendices



Validation Phase Testability Tasks



lation Phase Testability Tasks

TESTABILITY TASK COMPENDIUM
Task Reference Number VIA

PHASE: VALIDATION

FUNCTION: Testability Program Plan Preparation

TASK TITLE: Describe the testability program tasks

TASK OBJECTIVE: Define and describe the tailored tasks, to be pursued throughout the conduct of the Testability Program, as part of the Testability Program Plan which represents the overall testing strategy including functional test and on-line/off-line test considerations.

DESIGN DISCIPLINE TESTABILITY ($\bar{T}$) INTER-RELATIONSHIPS:

$\bar{T}$ INTERFACE	INPUT TO $\bar{T}$	OUTPUT FROM $\bar{T}$
● System Engineering	● Test requirements, preliminary system specifications, system tradeoff candidates	● $\bar{T}$ Task Definitions
● Design Engineering	● Test requirements, preliminary LRU specifications, tradeoff candidates	● $\bar{T}$ Task Definitions
● Maintainability Engineering	● Maintainability requirements	
● Logistics Support	● Logistic support concepts and constraints	
● Application Software	● Diagnostic software and BIT tradeoff candidates	
● Life Cycle Costing	● Baseline cost constraints	
● Program Management	● Program Master Schedule, Testability Program Plan Review/Critique/Approval	● Testability Program Plan

COST TRADEOFF INTER-RELATIONSHIPS: None directly applicable. The planned tasks will include provisions for cost estimating.

TASK SYNOPSIS:

1. Task Requirements.

1.1 The Testability Program Plan is a critical entry in the CDRL for an Advanced Development contract. The plan describes the contractor's understanding of the testability requirement and his approach to implementing and enforcing the requirement within his organizational structure and through his subcontractors. The Testability Program Plan must indicate that the contractor is giving adequate attention to supportability through logistic support analysis and testability analysis and that he is scheduling adequate development time to permit a testable design.(1)

1.2 The following excerpt from a candidate Data Item Description (DID) for a testability program plan contains the preparation instruction for a testability program plan.(1)

- The Testability Program Plan shall present the overall testing strategy including operational checks, periodic on-line tests, and off-line test considerations. It shall present milestones to be met to ensure that the final design achieves the required degree of testability. The plan includes the mechanisms for the reporting of progress, problems, and tradeoffs, and the enforcement of the proper use of testability design features by designers and subcontractors. The plan shall include the following:

  a. The work to be accomplished for each task delineated in MIL-STD-XXX.(3)

  b. Program milestones and customer reviews.

  c. The contractor organizational element responsible for the implementation of the Testability Program.

  d. Interfaces between that responsible organizational element and related elements such as:

     Systems engineering
     Design engineering
     Maintainability engineering
     Logistics
     Support equipment
     Training
     Operational software
     Diagnostic software
     Maintenance documentation
     Test

  e. Control over subcontractor and vendor testability program.

2. Implementation.

2.1 The plan will define how, during the validation phase, the following function will be performed:

a. Incorporate testability design

b. Perform BIT tradeoffs

c. Analyze inherent testability

d. Prepare testability analysis report

e. Prepare development specifications

f. Support PDR

2.2 The plan will define how, during the full scale development phase, the following tasks will be performed:

a. Perform test requirements analysis

b. Analyze design test effectiveness

c. Identify testability costs/penalties

d. Identify testability benefits

e. Prepare testability demonstration plan

f. Conduct testability demonstration

g. Prepare final testability analysis report

h. Support CI CDRS

i Evaluate operational testability

j. Support prime system CDR

2.3 The plan will define how, during the production phase, the following tasks will be performed:

a. Monitor production process and trends

b. Review change proposals

2.4 The plan will define how, during the operation & support phase, the contractor will monitor organizational/intermediate/depot testability data.

2.5 The Testability Program Plan will be contractually required by CDRL/DID. The degree of acceptability can be measured by:

a. Compliance with CDRL/DID requirements

b. Comparison to the topic checklist in the design requirements as a gauge of completeness

c. Contract data quality assurance function review of the completed plan for clarity, conciseness and editing principles.

3. Completion Criteria.

The task is completed upon Government acceptance of the Testability Program Plan.

TESTABILITY TASK COMPENDIUM
Task Reference Number V1B

PHASE: VALIDATION

FUNCTION: Testability Program Plan Preparation

TASK TITLE: Prepare testability program milestones

TASK OBJECTIVE: The Program Milestones show (1) the time phasing of each task and its interrelationship with other tasks, and (2) data submissions and their review, verification and utilization.

DESIGN DISCIPLINE TESTABILITY ($\bar{T}$) INTER-RELATIONSHIPS:

$\bar{T}$ INTERFACE	INPUT TO $\bar{T}$	OUTPUT FROM $\bar{T}$
● System Engineering	● Data Submissions, Reviews, Hardware Fabrication and Performance test schedules	● Testability Program Milestones, Schedule and Liaison
● Design Engineering	● Data Submissions, Reviews, Activity Schedules	● Testability Program Milestones, Schedule and Liaison
● Maintainability Engineering	● Data Submissions, Reviews, Activity Schedules	● Testability Program Milestones, Schedule and Liaison
● Reliability Engineering	● Data Submissions, Reviews, Activity Schedules	● Testability Program Milestones, Schedule and Liaison
● Application Software	● Data Submissions, Reviews, Activity Schedules	● Testability Program Milestones, Schedule and Liaison
● Life Cycle Costing	● Data Submissions, Reviews, Activity Schedules	● Testability Program Milestones, Schedule and Liaison
● Program Manager	● Data Submissions, Reviews, Activity Schedules	● Testability Program Milestones, Schedule, and Liaison

COST TRADEOFF INTER-RELATIONSHIPS: None directly applicable.

TASK SYNOPSIS:

1. Task Requirements.

1.1 Each milestone is a predetermined point of accomplishment which is clearly recognizable as an event which either does or does not occur at a predetermined point in time; for example: "Complete fabrication of developmental model." Areas or phases known to be potentially controlling efforts or those known to be pushing the state of the art should be carefully identified and milestoned.

2. Implementation.

2.1 The Testability Milestone Plan consists of a series of clearly defined milestones with the scheduled (planned) time span and completion date of each. The Testability Milestone Plan format and symbology should be standardized and consistent with other plans within the program. Key milestone data for the parent program should also be displayed to show the interrelationships with Testability activities.

2.2 The Testability Milestone Plan should contain the testability function schedules. The following is a list of testability activities that should be considered in formulating the milestones.

a. Validation Phase.

- Design Units to be compatible with selected or available ATE
- Select parts for $\overline{T}$; provide direct designs and test sequences
- Design Units with testable physical partitioning
- Incorporate initialization into design
- Incorporate controllability into design
- Incorporate observability into design
- Determine the number and placement of UUT test points
- Incorporate system level BIT
- Incorporate general $\overline{T}$ design features
- Tradeoff alternate designs of a BIT, external automatic test, and manual test mix
- Characterize failure modes (types)
- Estimate frequency of occurrence of each failure mode
- Show that qualitative $\overline{T}$ is included in design
- Prepare $\overline{T}$ analysis model for each preliminary design
- Define Figures-of-Merit for inherent $\overline{T}$

- Analyze HW/SW BIT features
- Prepare $\overline{T}$ analysis report
- Prepare CI development specifications
- Prepare Computer Program CI Development Specifications (CPCI)
- Prepare ETE specifications <u>or</u> select ETE
- Support PDR

b. Full Scale Development Phase.

- Produce a Test Requirements document for each UUT
- Produce a Diagnostic Software Specification
- Insure total testability
- Predict levels of fault detection and fault isolation for the equipment and each UUT
- Identify development costs and recurring costs and penalties
- Estimate testability impact upon development/operation and support
- Prepare Testability Demonstration Plan
- Conduct Testability Demonstration
- Prepare Final Testability Report
- Review testability features and predicted testability parameters
- Monitor/evaluate/propose corrective action
- Provide field testability data

c. Production Phase.

- Monitor production process and trends. Review change proposals

d. Operation and Support Phase.

- Monitor Organizational, Intermediate, and Depot level $\overline{T}$ data

e. Contractual dates such as contract award, contract end, and program reviews

3. Completion Criteria.

3.1 Task is completed on government acceptance of the Testability Program Plan.

TESTABILITY TASK COMPENDIUM

Task Reference Number V1C

PHASE: VALIDATION

FUNCTION: Testability Program Plan Preparation

TASK TITLE: Identify the testability responsibilities, authority and interfaces

TASK OBJECTIVE: Identify the organizational responsibilities and authority for testability management including control of subcontracted engineering, levels of control for design requirements, reviews and documentation. Describe interfaces between the organizational element responsible for testability and other related elements. Show how adequate surveillance will be maintained to enforce all testability requirements.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
• Program Manager	• Testability Tasks, Subcontract Engineering Control, Design Requirements Control, Review Control, Documentation Control	• Documented Acknowledgement of Authority and Statements of Responsibilities
• Customer	• Division of Responsibility and Authority	• Documented Acknowledgement of Authority and Statements of Responsibilities

COST TRADEOFF INTER-RELATIONSHIPS: None

TASK SYNOPSIS:

1. **Task Requirements.**

1.1 The following is a list of testability organizational interfaces.

a. System Engineering
b. Design Engineering
c. Maintainability Engineering
d. Logistics Support
e. Support and Test Equipment

f. Training

g. Application Software

h. Maintenance Documentation

i. Production Engineering

j. Reliability Engineering

k. Life Cycle Costing

l. Safety Engineering

m. Program Management

n. Test Engineering

o. Human Factors

2. Implementation.

2.1 In all interfaces, the testability engineer should be prompt, active and assertive. Testability is most effectively instilled into a system if design guidelines are provided when design commences. It is inefficient for the testability (or any other -ility) engineer to take the role of a post-design critic and difficult as well as uneconomical for designers to incorporate changes to documented designs when those changes reflect design criteria which could have and should have been disclosed during or ahead of design formulation. Timeliness is therefore a principal responsibility for the testability engineer.

2.2 The testability engineer's charter is to establish and maintain an effective testability program that is an integral part of the overall design effort. In order to accomplish this he must form a close liaison with the other design disciplines. The "Design Discipline Testability $(\bar{T})$ Inter-Relationships" section of each compendium is a suggested list of groups that the testability engineer should coordinate with for that particular task.

2.3 Authorities to be established typically include:

- Imposition of qualitative and quantitative testability requirements into design goals and into CI specifications.
- Imposition of quantitative testability requirements into BIT design and ATE or ETE selection.
- Imposition of testability requirements on software development.

2.4 Responsibilities to be accepted by testability engineering typically include:

- Timely delivery of inputs to design goals and CI specification documents.
- Timely delivery of inputs for BIT design and ATE or ETE selection and thorough review of BIT design and ATE or ETE selection tradeoffs.
- Timely delivery of inputs for software development and thorough review of program lists for testability.

2.5 The Test Program Plan will be contractually required by CDRL/DID. The degree of acceptability can be measured by:

a. Compliance with CDRL/DID requirements.

b. Thoroughness in establishment of meaningful assignments of authority and responsibility for each testability task.

c. Thorough documentation of organizational interfaces.

3. Completion Criteria.

The task is complete upon acceptance by the government of the Testability Program Plan.

TESTABILITY TASK COMPENDIUM

Task Reference Number V2A

PHASE: VALIDATION

FUNCTION: System Design $\overline{T}$ Incorporation

TASK TITLE: General Testability Design Guidance

TASK OBJECTIVE: This task is somewhat hypothetical in that it provides a vehicle for listing of a generalized set of design guidelines for early transmittal to the designers.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering		● $\overline{T}$ Design Guidelines
● Safety Engineering		● $\overline{T}$ Design Guidelines
● Design Engineering		● $\overline{T}$ Design Guidelines
● Reliability Engineering		● $\overline{T}$ Design Guidelines
● Maintainability Engineering		● $\overline{T}$ Design Guidelines
● Logistics Engineering		● $\overline{T}$ Design Guidelines

COST TRADEOFF INTER-RELATIONSHIPS: Providing early guidance to the designers minimizes the cost of design by avoiding re-design and/or subsequent engineering changes.

TASK SYNOPSIS:

1. **Task Requirements.**

This task is a vehicle for providing early guidance of a general nature to the designers. It is aimed primarily at circuit card level, but the principles are applicable at higher levels of the hardware indenture. Tasks V2B through V2I provide more specialized guidance related to specific aspects of testability and in fact repeat some parts of the guidance of this task. The guidelines should also be provided for information to engineering disciplines other than design.

2. Implementation.

The appendix hereto is a separable set of guidelines which may be directly provided to the designer.

3. Completion Criteria.

The task is complete upon completion of the design effort with all realizable guidance points contained in the design.

APPENDIX V2A1(2)

Testable Design

The general characteristics of a testable design are:

- Test control of internal nodes, including initialization
- Observability of internal nodes, directly or inferable
- Mechanical and electrical compatibility with available ETE/ATE
- Functional partitioning
- Conservative timing and signal tolerances
- Well-behaved failure modes
- Restricted fan-out count
- Simple, straightforward, regular designs
- Support of testing at several levels of hardware indenture

Test Software (3)

- Device test programs should be programmed in an ATE high-order source language, e.g., ATLAS or OPAL.
- Source programs should be defined in independent blocks traceable to a functional grouping of device test (UUT) requirements.
- The blocks should be annotated to describe their test functions.
- Block size should be restricted to test sequence size (e.g., fifty test measurements) and also run-to-completion-time (e.g., five minutes).
- The test blocks should contain a common set of language procedures (subroutines and data base) wherever possible.
- The test block executions should be individually initiated by and terminated into the test executive software.

PCB Electrical Layout

- Standardize pin location of common signals such as power, ground, clock, clock return for a card family. Power pins should be located on the connector ends (board edges).
- Design the system to use only one voltage if at all feasible in order to avoid potential damage that may result from transients or improper voltage sequences.
- If more than one voltage is required, segregate multiple voltages to as few card types as possible. Provide adequate separation to prevent accidental shorting with test probes.
- Use a single logic family for the logic design. If more than one logic family must be used, select ones with common power requirements and pin assignments, and common input/output signal compatibility and pin assignments.
- If logic families with a signal interface problem must be used, partition the families so they can be tested individually.
- Select logic devices that are independent of specific clock rates, controlled rise and fall times, and/or specific gate propagation delays.
- Run a complete characterization test study on new type devices to avoid sneaky soft (come and go) problems. Be conservative in designs that use such parts, especially with respect to timing parameters. This rule applies with strong emphasis to complex, large-scale devices.
- Design each card to be a functionally separate package. Otherwise separate multiple functions on a card by partitioning so that each function can be tested separately.
- Subdivide large logic boards (>100 ICs) into smaller sections for easier testing. Separate the Vcc paths or use tri-state logic to allow isolating one section from the others for testing.
- Do not mix digital and analog circuitry on the same card if feasible. This does not include simple timing circuits.
- If both digital and analog circuitry must be included on the same board, partition the board into digital and analog portions so that each can be tested separately. If the circuit design permits, provide the digital/analog interconnections externally via I/O pins.
- Design circuits that do not require adjustments. If adjustments must be incorporated, design the board so that the adjustment can be made and locked during test.

- Design I/O lines to allow for inevitable shorting by providing current limiting or crowbar protection techniques.

- Design I/O lines and test points to accommodate capacitive loads larger than intended in the system. This enables the PCB to be accommodated by a wider range of test adapters and ATE sets.

- Provide means for external control of each node independent of the associated logic by providing direct access via an I/O pin or a test point. This allows external control of the node for troubleshooting purposes.

- Provide sufficient access to the internal circuit by bringing internal test and control nodes to unused pins on the connector, or to pins on the test point connector.

- Provide simple means for initializing (setting to an initial state) all storage elements, using signals applied to I/O pins or test points. A single direct reset is preferred; a short (≤16 bit) count sequence is acceptable.

- Make display refresh circuitry capable of being disabled, with the refresh cycle then being capable of being controlled by the tester.

- Incorporate built-in-test capability into very large logic boards such that BIT can be initiated by the ATE and the test results can be evaluated by the ATE.

- If access to internal circuitry of a PCB is limited, provide BIT capability on the card that can be initiated and read by the ATE.

- Use display LEDs on the PCB to indicate proper operator of important circuits. Examples are: power supply voltage is O.K., clock is present, a phase-locked loop is locked, etc.

- Use position indication fault indicators and displays such that a good test always results in an "ON" condition. A defective indicator or display then always indicates a fault condition, either due to the input or of itself (if it is faulty).

- For a critical display, provide an alternate method of testing such as "push-to-test" to provide positive verification of its operation.

Current Requirements and Limitations

- Limit input signal requirements including clock and master clear to ≤20ma. Provide on-board buffering if more drive capability is required.

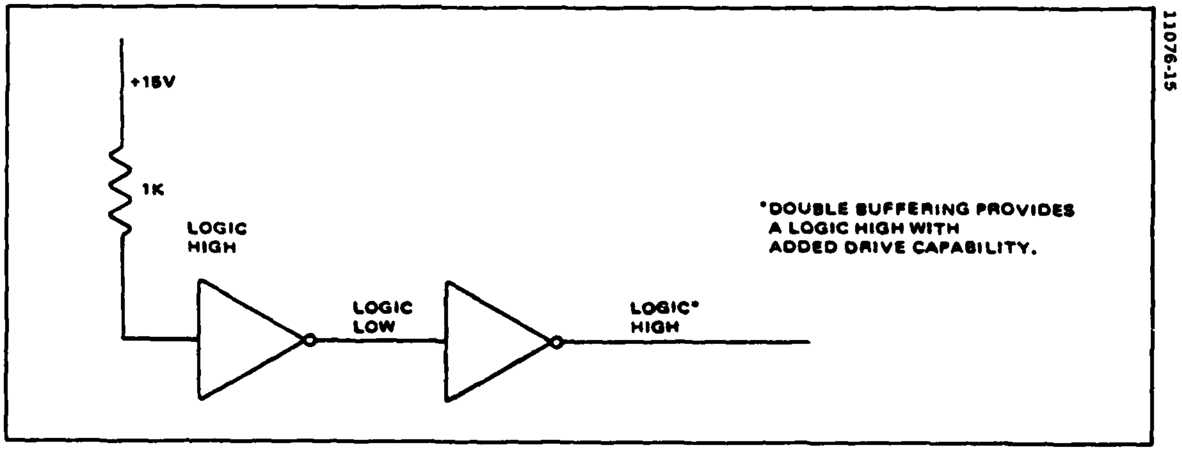
- Buffer clock inputs before they fan out to avoid loading down clock drive outputs.

- Provide sufficient current capability in high frequency and pulse outputs to drive a low impedance transmission line.

- Provide current limiting on output lines where there is a likelihood of a cascading of failures, should a single component fail.

- Do not tie direct set or reset lines to Vcc or ground. Decouple Vcc through a resistor to provide a logic high; pass a decoupled high through an inverter to provide a logic low. This protects against shorting Vcc to ground due to a component failure. (See figure 1.)

- Alternatively use a grounded input inverter to provide a logic high, and a second inverter driven by the first to provide a logic low. This greatly aids in locating a fault due to a shorted fixed "zero" or fixed "one" input.



*DOUBLE BUFFERING PROVIDES A LOGIC HIGH WITH ADDED DRIVE CAPABILITY.

Figure 1. Isolate Logic Highs and Lows From Power Buses

- In some cases where real estate is a problem, a logic signal with a diode input can be safely tied to Vcc.

- Design I/O lines to allow for inevitable shorting by providing current limiting or crowbar protection techniques.

- Design I/O lines and test points to accommodate capacitive loads larger than intended in the system. This enables the PCB to be accommodated by a wider range of test adapters and ATE sets.

- Provide means for external control of each node independent of the associated logic by providing direct access via an I/O pin or a test point. This allows external control of the node for troubleshooting purposes.

- Provide sufficient access to the internal circuit by bringing internal test and control nodes to unused pins on the connector, or to pins on the test point connector.

- Provide simple means for initializing (setting to an initial state) all storage elements, using signals applied to I/O pins or test points. A single direct reset is preferred; a short ($\leq$16 bit) count sequence is acceptable.

- Make display refresh circuitry capable of being disabled, with the refresh cycle then being capable of being controlled by the tester.

- Incorporate built-in-test capability into very large logic boards such that BIT can be initiated by the ATE and the test results can be evaluated by the ATE.

- If access to internal circuitry of a PCB is limited, provide BIT capability on the card that can be initiated and read by the ATE.

- Use display LEDs on the PCB to indicate proper operator of important circuits. Examples are: power supply voltage is O.K., clock is present, a phase-locked loop is locked, etc.

- Use position indication fault indicators and displays such that a good test always results in an "ON" condition. A defective indicator or display then always indicates a fault condition, either due to the input or of itself (if it is faulty).

- For a critical display, provide an alternate method of testing such as "push-to-test" to provide positive verification of its operation.

Current Requirements and Limitations

- Limit input signal requirements including clock and master clear to ≤20ma. Provide on-board buffering if more drive capability is required.

- Buffer clock inputs before they fan out to avoid loading down clock drive outputs.

- Provide sufficient current capability in high frequency and pulse outputs to drive a low impedance transmission line.

- Provide current limiting on output lines where there is a likelihood of a cascading of failures, should a single component fail.

- Do not tie direct set or reset lines to Vcc or ground. Decouple Vcc through a resistor to provide a logic high; pass a decoupled high through an inverter to provide a logic low. This protects against shorting Vcc to ground due to a component failure. (See figure 1.)

- Alternatively use a grounded input inverter to provide a logic high, and a second inverter driven by the first to provide a logic low. This greatly aids in locating a fault due to a shorted fixed "zero" or fixed "one" input.

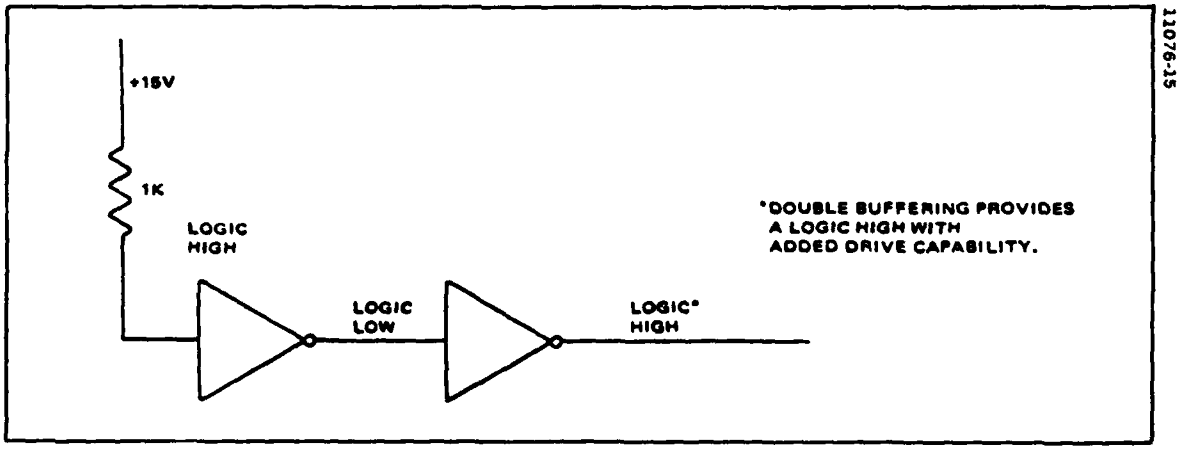


Figure 1. Isolate Logic Highs and Lows From Power Buses

- In some cases where real estate is a problem, a logic signal with a diode input can be safely tied to Vcc.

Redundant Circuitry

- Avoid logically redundant circuits. A connection in a circuit is said to be redundant if no change occurs in the output of the circuit when the connection is open. (See figure 2)

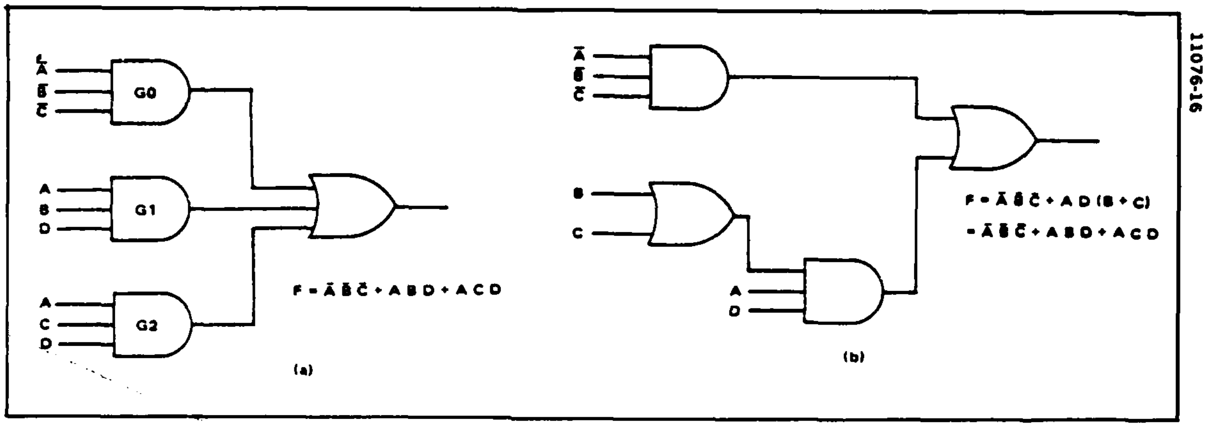


Figure 2. Provide Test Access by Redesign to Avoid Fault Redundancy

Both circuits generate the same logic function. However circuit (a) gives the same output even when the B input on gate G1 is stuck at 1, or the C input on gate G2 is stuck at 1. In either case the output becomes F = $\overline{ABC}$+AD, hence the fault cannot be isolated to a specific gate.

With either B or C stuck at one, circuit (b) output also becomes F = $\overline{ABC}$+AD, but now the fault can be isolated to the $B \cdot C$ gate.

- Logically redundant circuits can help solve race problems by providing overlapping control at a clock time.

- Redundant design can prevent a glitch. (See figure 3)

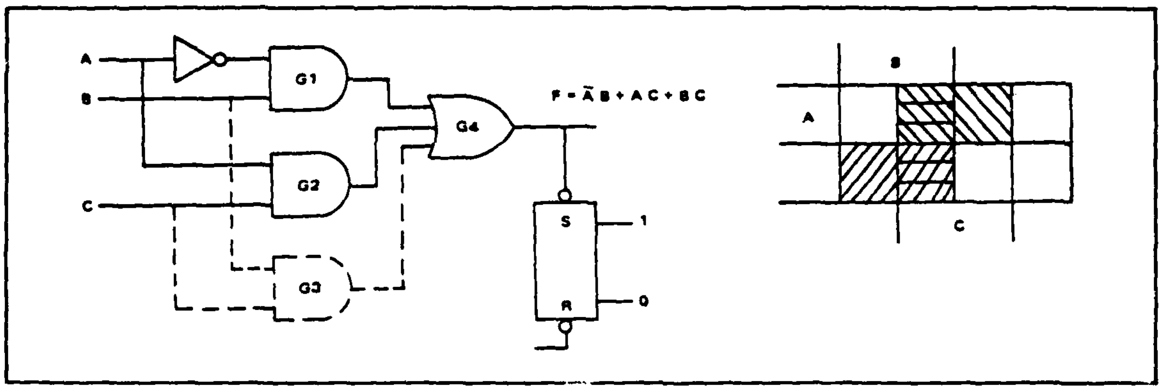


Figure 3. Gate G3 is Redundant, as can be seen on the Veitch Diagram

Without G3, a glitch could occur at F during the crossover time when gate G2 (A˙C) releases control and gate G1 (A˙B) takes control (or vice versa). The glitch could direct set the flip-flop. By adding the redundant gate G3 (B˙C), the glitch cannot occur.

Wired AND and Wired OR Connections

- Do not use wired AND or OR because of the ambiguity that is created in trying to localize a fault to the specific faulty gate.
- Do not use wired AND (OR) with high powered TTL or Schottky TTL; they cannot be direct driven low safely for more than 1 second.
- Break wired AND connections into smaller ambiguity groups as illustrated. (See figure 4) The before case shows six gates were ANDed; the after case shows three pairs of wired ANDed gates. The ambiguity factor has been reduced from 6 to 3.

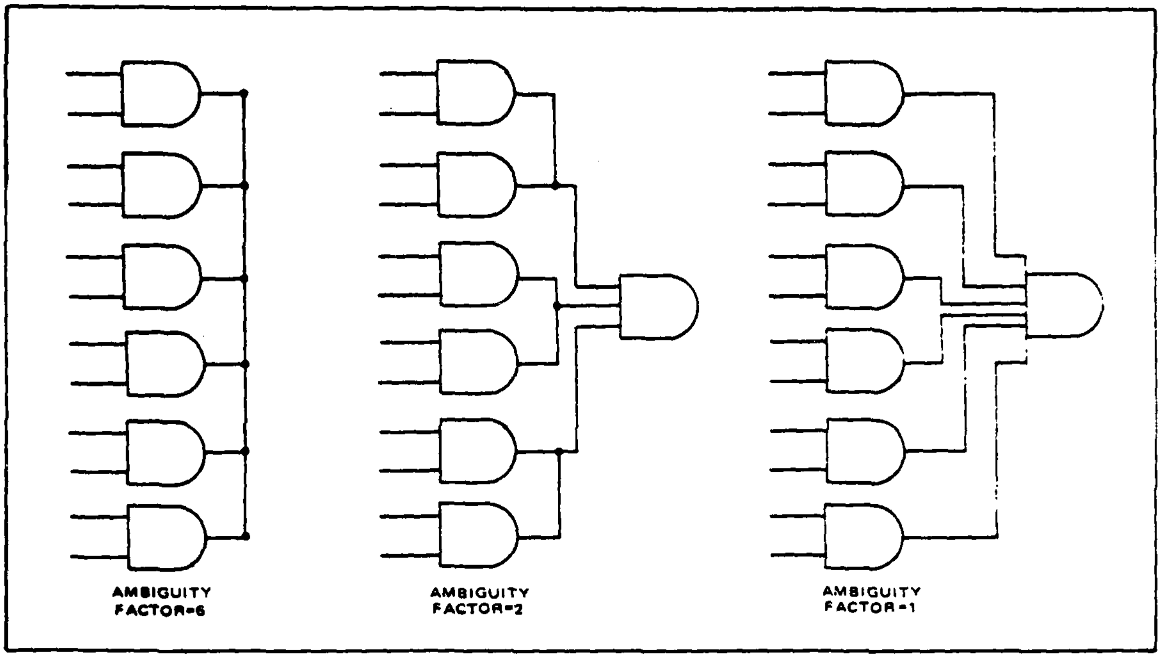


Figure 4. Redesign Gate Networks to Reduce Fault Ambiguity Factor

- Use wired AND (OR) for external control connections as a means of injecting test stimuli into card under test. (See figure 5.)

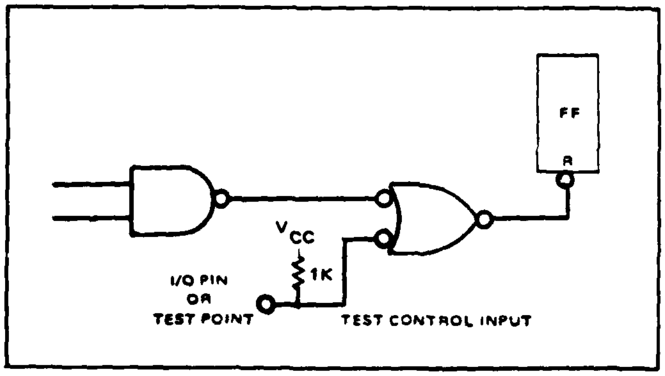


Figure 5. Wire OR for Test Control Input

Internal Clocks and Counter Chains

- Use synchronous clocking systems only. This reduces potential timing hazards to narrow and easily identifiable areas of logic. With synchronous clocking the typical digital logic assembly can be tested on a static digital test system with less concern about timing response.

- Provide an easy means of disabling or bypassing on-board clocks (oscillators, etc.) so that the necessary clock stimuli can be provided from the test equipment. (See figure 6.)

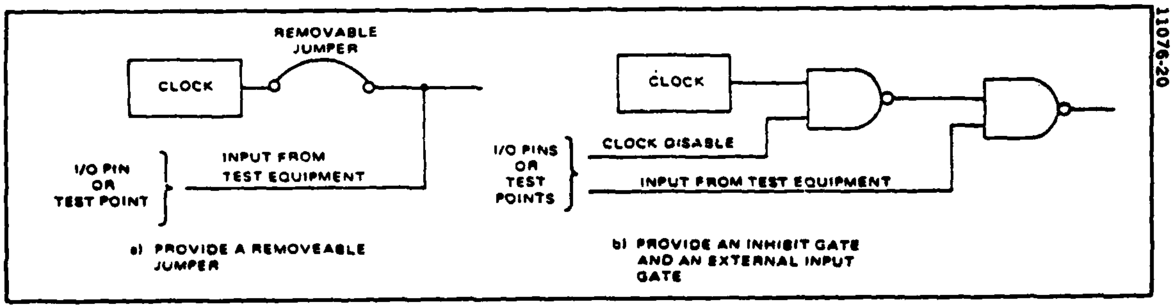
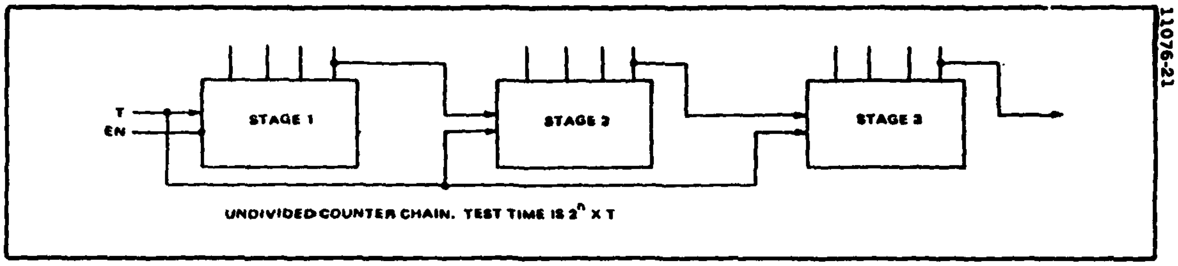


Figure 6. Techniques for Disabling On-Board Clocks to Allow Off-Board Clocking

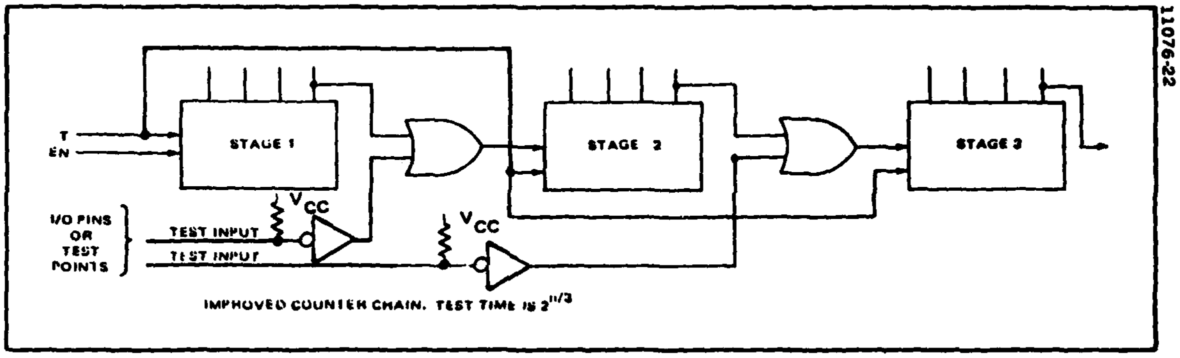
- Provide means for easily inhibiting an on-board oscillator (putting it into a non-oscillatory mode).

- Break up long counter chains into smaller segments, preferably of equal length, such that the total count time provides a reasonable total test time. (See figures 7, 8 and 9.)



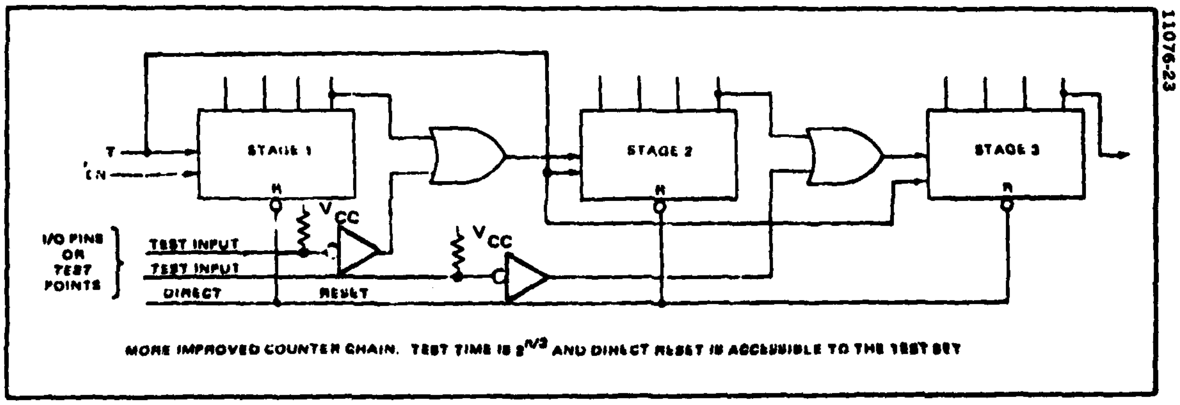
UNDIVIDED COUNTER CHAIN. TEST TIME IS $2^n \times T$

Figure 7.



IMPROVED COUNTER CHAIN. TEST TIME IS $2^{n/3}$

Figure 8.



MORE IMPROVED COUNTER CHAIN. TEST TIME IS $2^{n/3}$ AND DIRECT RESET IS ACCESSIBLE TO THE TEST SET

Figure 9.

Sequential Circuits

- Initialize all sequential circuits to a known start condition utilizing the shortest possible sequence, ideally one transition, and never more than 20 transitions.

- Provide means to initialize all sequential logic elements from I/O pins or test points. This applies to flip-flops, counters, registers, memories, etc. (See figure 10.)

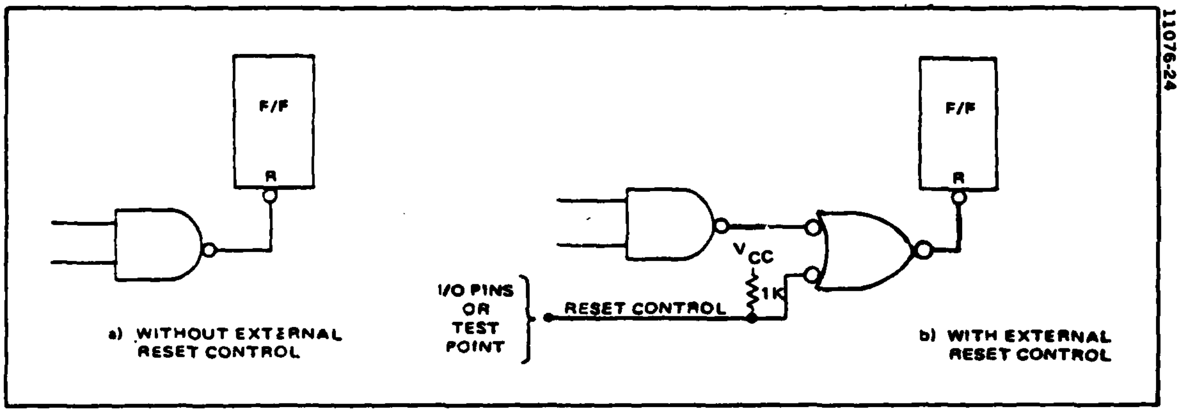


Figure 10. Flip-Flop Modified for External Reset Control

- If an I/O pin or test point is not available, provide initialization by means of "power-up." (See figure 11.)

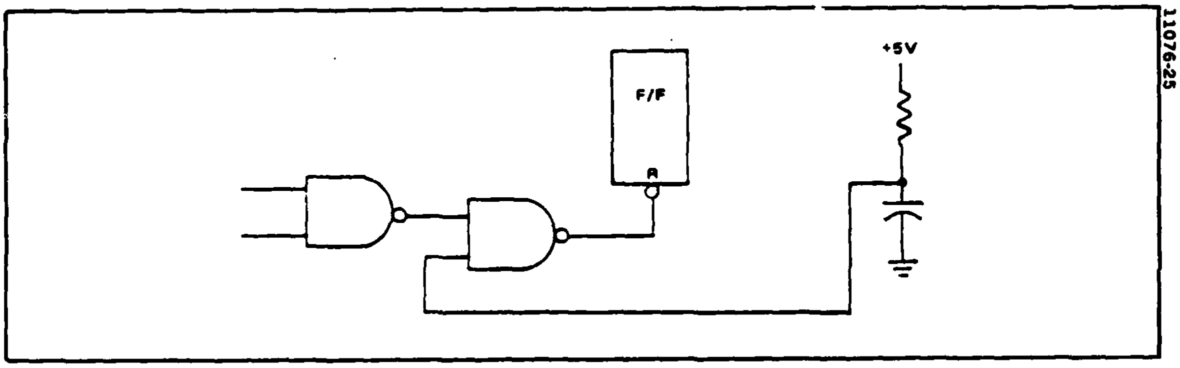


Figure 11. Power-Up Reset

- If unused direct reset and set pins must be tied down, always do so through a decoupling resistor. This can make these inputs available for test purposes, and prevents shorting the power bus through a bad chip.

- Avoid arbitrary use of edge-triggered devices such as D flip-flops and J-K flip-flops. They increase modeling requirements, yield lower test quality in nodal-fault testing, and often exhibit race conditions.

- Provide access for test control to internal nodes of complex sequential circuits by means of gates, I/O pins, and/or test points. (See figure 12.)

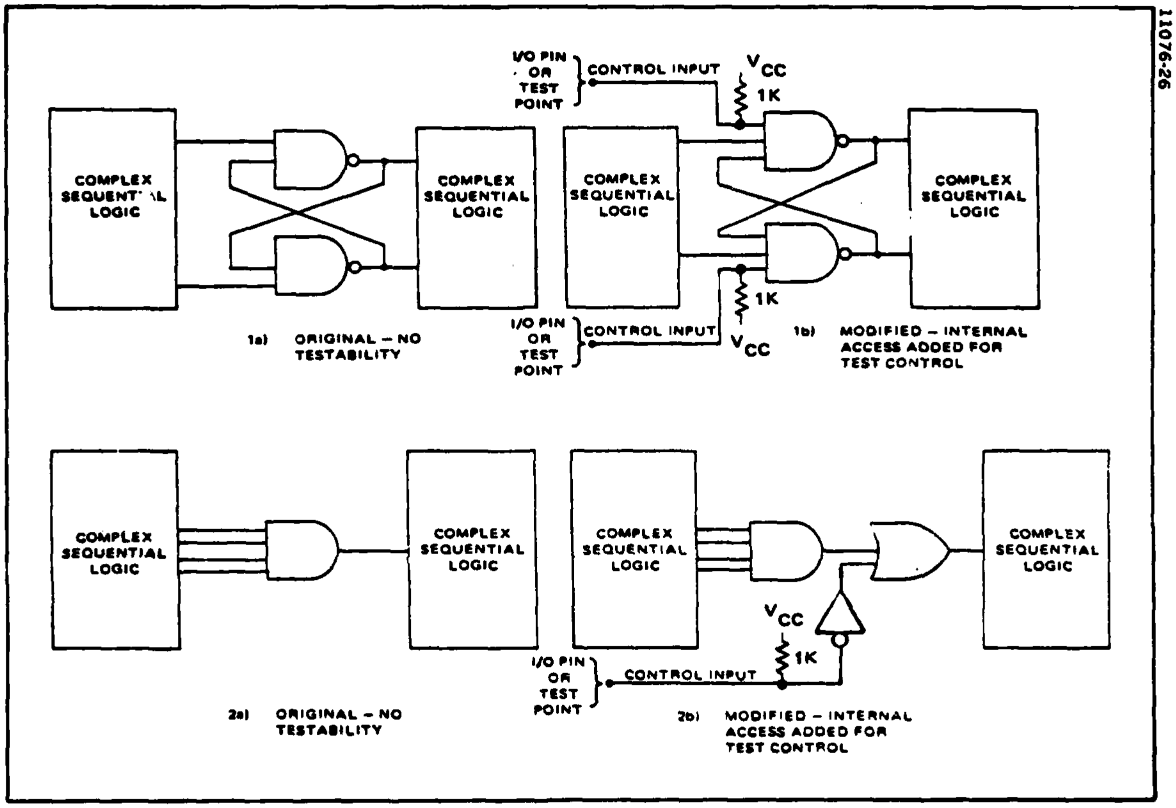


Figure 12. Provide Access to Internal Modes for Test Control

- Provide positive controls to prevent logic lockups that can be cleared only by a master clear, or indeterminate outputs such as can occur from an R/S flip-flop.

- Logic designers should provide state diagrams for sequential logic circuits. This will enable the test engineer to analyze tests from valid sequences only, and to ensure that initialization prevents the occurrence of invalid sequences and lockup.

Asynchronous Circuits, Monostables (One-Shots) and Feedback Networks

- Do not use asynchronous clocking. It typically requires use of edge-triggered flip-flops, monostables (one-shots) and/or delay elements to control the functional sequence. Testing monostables and delay lines usually requires manual test generation for accurate timing tests, voiding the use of automatic test generation.

- Asynchronous clocking is highly undesirable because it usually results in the use of various asynchronous clocking methods distributed throughout a functional module. This, in turn, establishes a need for a multitude of separate test environments and extensive costly manual test generation.

- Do not use asynchronous circuits. If they must be used, provide means for synchronizing them in test, or provide means so that they can be tested by themselves.

- Do not use monostables on digital logic boards.

- If monostables must be used, they must be capable of producing synchronized, predictable, and repeatable pulses.

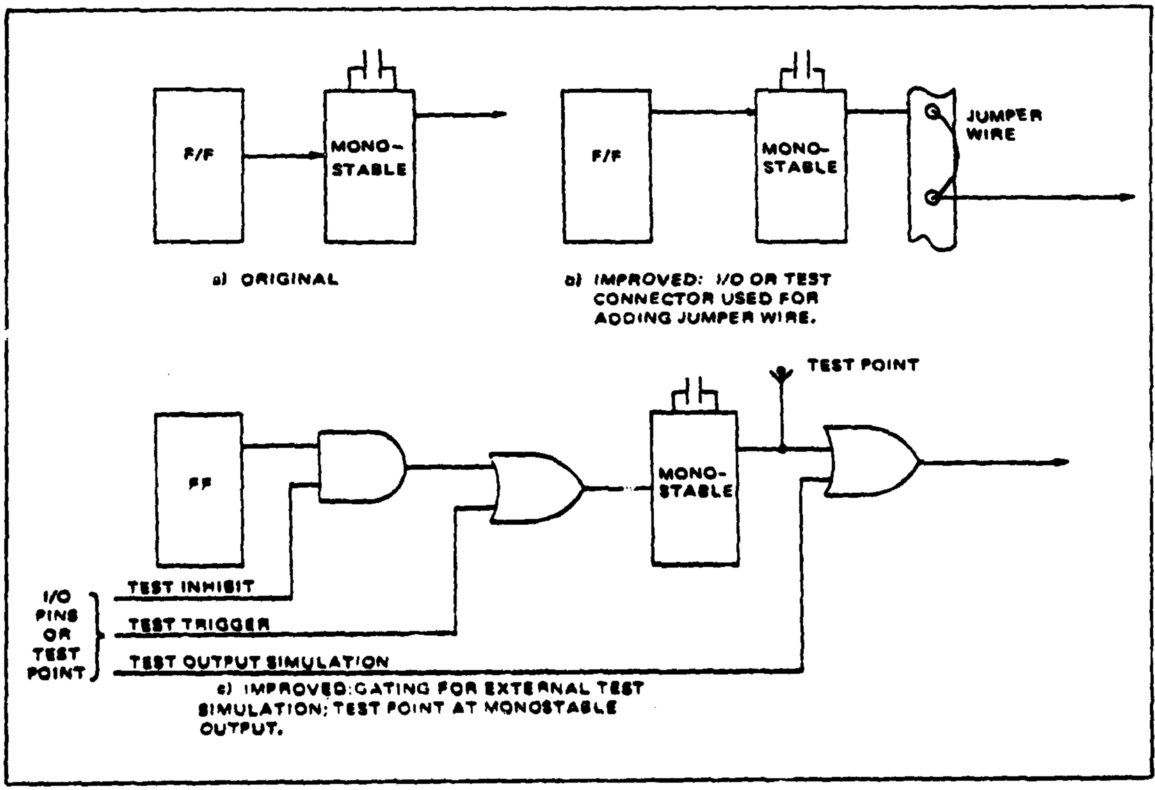


Figure 13. Electrical Disconnection of Monostables with External Simulation Path Provided

- If use of a monostable is mandatory, provide means of electrically disconnecting the circuit input and output by gating or by jumpers. Provide a path for external test stimulation of the monostable and a test point to measure the monostable output. (See figure 13.)

- Do not use asynchronous feedback networks. Particularly avoid those dependent upon propagation delay. Replace such a circuit with a synchronous feedback function.

- Provide means of breaking all feedback paths so that faults in the feedback loop can be isolated. Ensure that the circuit can be operated in the open loop mode by providing an alternate test input at the place where the loop is opened. Use a jumper wire or gating to open the feedback path. (See figure 14.)

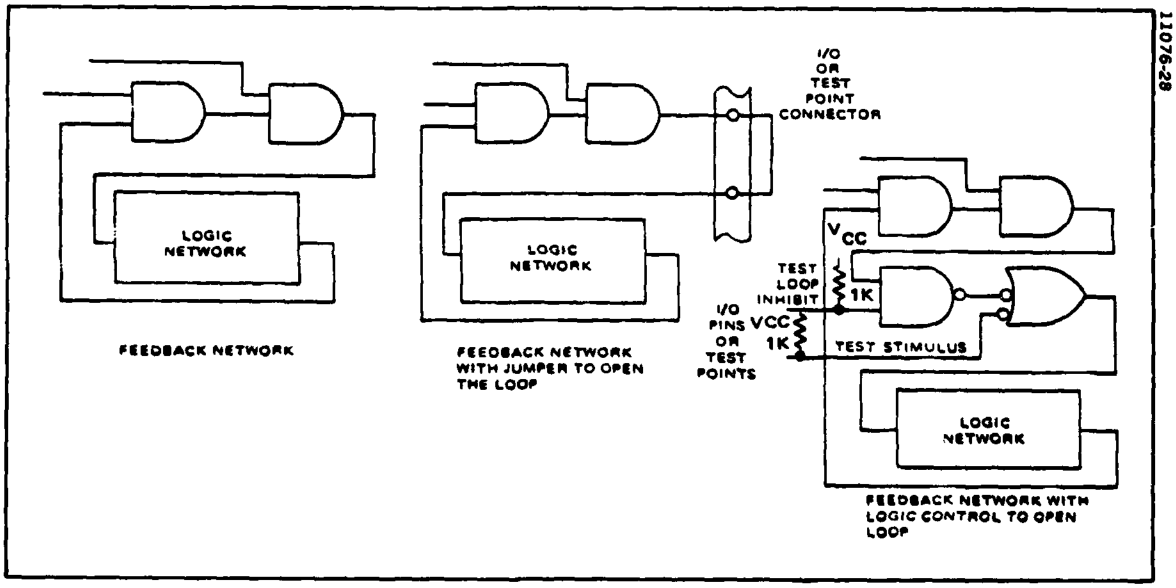


Figure 14. Break Feedback Paths

Large Scale Rules

The large scale testability design rules are those that apply particularly to LSI and VLSI devices including microprocessors, microprocessor support chips, RAMs and ROMs, UARTs, etc.

Some of the rules already discussed apply in general to large scale devices, when appropriately modified to meet the special conditions applicable to such devices. The rules below are those particularly applicable to design using large scale devices.

- The card designer and the test engineer must both have detailed knowledge of large scale components and the problems of interfacing between them. This knowledge is required to develop a viable design that includes the testability needed to detect faulty operation and to isolate the fault to a defective component.

- Test each lot of an LSI device to ensure that the device characteristics have not changed. A different lot number of a large scale device, particularly in its early life, can mean differences in characteristics.

- Characterization studies to establish worst case limits of critical parameters are essential for LSI devices. Semiconductor manufacturers generally do not provide information on test vector patterns that best test their devices: They generally only furnish DC and timing characteristics. User experience is that characterization studies are worth their weight in gold.

- Use dynamic devices (dynamic RAM, microprocessors, etc.) only when absolutely required. Provide separate access to the refresh clock at an I/O pin or test point.

- The initialization rules also apply specifically to volatile memory elements. Conditional initialization is not acceptable without a controlling override absolute initialization. A predetermined short sequence of clock counts to initialize is acceptable.

- Use partitioning judiciously to separate the functioning elements of a product to improve overall testability characteristics.

- Avoid concentrating many LSI chips on a single PC board. The resulting structure is difficult to test and troubleshoot because of the restricted physical access and the restricted test visibility.

- Utilize chips with power ON/OFF capability to electrically partition groups of chips for independent testing of parts of the PCB one at a time. Then the parts can be combined piecewise or en masse, depending upon interface complexity. This approach greatly simplifies fault localization and applies the proven concept of bottom-up integration PCB testing.

- After checking specification requirements, if possible, mount all large and/or complex ICs in sockets so that they can be removed to allow testing by partitioning.

- When doing characterization studies of an LSI device, know what phase in the life cycle the device has achieved. These phases are as follows:

  - Characterization Phase-tests define the critical characteristics and characteristic limits. Processes and parameters are subject to change as the device matures. Production is low.

  - Prematurity Phase- production is increasing and processes and parameters are more stabilized.

  - Shell Phase- production growth is rapid. Tests are pared to a "shell" of critical requirements and some testing of critical processes. Dedicated test equipment is selected. Competition in production is apparent.

  - Maturity Phase- production has peaked. Dedicated test equipment is used for stationized testing of the "shell" of tests selected in the previous phase. Process control testing is minimal but adequate. Competition is reaching its peak.

  - Post Maturity Phase- the device is turned over to the sustaining production group. Routine testing, based on past learning, is done at minimal cost but with tight control of processes. Prices are fixed. Competition has decreased but also has stabilized.

Microprocessor and Support Chip Rules

- Capture the microprocessor design effort for later analysis and use in the test development effort. Include all data derived from the use of design and debugging aids such as an MDA (microprocessor development aid), logic analyzer and/or other design tools, including complete information on all equipment setups, procedures, and results. Use computerized tools to perform this data capture automatically and painlessly.

- In a microprocessor system the functional operating characteristics of the circuit are not necessarily associated with specific hardware components.

- Microprocessor systems generally lend themselves well to ordered partitioning. The average microprocessor system has six unique chips.

- Due to their complexity, microprocessors are very difficult to test when incorporated into a PCB as part of the logic design. If they can be mounted in sockets, this permits removing them for testing separate from the rest of the PCB.

- Use microprocessors from the same family in a multiprocessor configuration, rather than different microprocessors unless there are specific compelling reasons to do so. Multiprocessor configurations developed around a microprocessor family are much easier to test.

- Verify that the microprocessor can be run in the tri-state mode (high impedance state).

- Provide inhibit gates for microprocessor outputs that cannot be put into the tri-state high impedance state.

- Separate bit-slice microprocessors into elementary slices for test control. This permits the use of a much simpler program to test each elementary slice independently. Interslice coupling can then be tested by a relatively simple test program.

- Tie unused CPU control inputs to power buses only through resistors or inverters, and provide access through I/O pins or test points. This provides access to the control line for test purposes and prevents shorting of the power bus if the device is defective.

- Microprocessor phase clocks must conform strictly to the requirements of the microprocessor and its associated ships, rather than being dependent upon any other clock circuitry.

- Provide a means to loop I/O port outputs back to inputs for a very effective I/O test. This has been implemented using an on-board test connector together with a test adaptor connector at the board edge.

- Test the clock circuit immediately after POWER UP tests. An absent, jittery, or noisy clock can play havoc with the rest of the hardware functions. Abberations of the clock signal may prove fatal, and marginal operation often leads to hard-to-find transient problems.

- Use progressive testing to verify the hardware, employing simple programs. The best test sequence is Power, Clock, ROM, RAM, through test of I/O ports and interrupts, and bus control logic.

- Test non-CPU circuitry independently by tri-stating the CPU and utilizing the inhibit gates to isolate the CPU from the other circuitry.

- Perform passive preliminary tests to be sure that the board is free for communication.

- The three widely used microprocessor test methods are:

  1. Actual use
  2. Stored response from a known good board
  3. Algorithmic pattern generation

- Subdivide the microprocessor into functional units (adder, multipler, incrementer/decrementer, program counter, input buffer, output buffer, stack pointer, register address, etc.). Devise a diagnostic fault detection only test for each of these units, since they cannot be repaired. Utilize the principle of "expanding on the kernel" in devising the sequence of these tests.

- The principles of individual functional subunit testing and expanding-on-the-kernel provides a controlled progressive means of developing programs. It keeps one from being overwhelmed by the complexity of the device.

- Devise a test for the microprocessor kernel, the minimum configuration of microprocessor and ROM needed to run a very simple test program. Use this proven part to expand testing to an untested area. Continue this test expansion process until the complete microcomputer system has been tested.

- Properly designed-in testability can significantly simplify testing of microcomputers. Three reasons support this conclusion: (1) resident self-test programs written in the microcomputer's own language can be stored directly on ROM. By "expanding on the kernel" these programs provide excellent means for verification of operation, and can easily generate ample stimulus for fault diagnosis. The same self-test can be used for proving out the design in the laboratory, testing in production and servicing in the field; (2) the system bus, which is centrally located and connects to many components in the system, can be physically designed to provide easy access for application of test stimuli and visibility of test results; (3) bus oriented architectures are inherently readily designed into easily diagnosable modules.

- All software storage locations that are accessed by the CPU must be initialized by an initializing routine that precedes the test routine proper. Include in the initializing sequence RAM locations, peripheral IC registers, input ports, CPU registers and PIDs, etc., that are not affected by hardware initialization. The test designer must be aware of exactly what registers and addresses are saved on the stack at the initiation of an interrupt or subroutine call. When the return is executed these saved registers are read back from the stack into the processor locations, etc., from which they were saved. The initialization procedure must precede any interrupt action or subroutine call to ensure that proper control of these registers, etc., is maintained at all times.

- Microcomputer design should include a built-in self-test, which requires about 1 K-byte of ROM. A software sequence, external switch, or test point can be used for the initiating maintenance action of this test.

- An alternative to built-in test is to utilize an external signal, switch-action, or connector to overlay a self-test program onto the applications program RAM memory space, and operate the test from there.

- Write general purpose self-test modules that apply to any system based on a particular microprocessor; for example, a CPU test module and a RAM test module. Usually only small changes are required to adapt the test modules from one system to another.

- Write I/O driver routines in applications programs as subroutines so that they can also be incorporated into test routines as a subroutine call. This saves both code space and program development time.

- The first step to localize a microprocessor system malfunction is to determine its generic source. Is it caused by hardware, firmware or software?

Buses

- The first step in functional board test of a microcomputer board is to make sure the bus structure is free of manufacturing faults. This should be done even when the board has been through in-circuit component inspection test (ICCIT).

- Provide access to microcomputer, RAM, ROM (or other device) address, data, and control buses as applicable. This can be done in a number of ways.

  - The most flexible access is to bring all buses to I/O connector pins.

  - Route buses that do not have I/O access to test points (a test connector).

  - Provide space to access the buses by means of a dip-clip. (If the boards are to be conformally coated this is not useful for post-delivery factory retest or for field testing).

  - Provide access to the control line for tri-state buses at I/O pins or test points.

- The rule for putting pull-up resistors on the same board as their driving device applies particularly to tri-state devices.

- Add a default bus driver to actively pull the bus to a known state when no other bus drive is active. Decode hardware, using address and bus control inputs, is used to generate the default bus control.

- Provide a means for floating or opening each bus line to assist in fault isolation and to follow free running operations. Use one or a combination of the following methods:

  - Hardware or software controlled chip select lines, jumper wires, shorting plugs, socketed components.

  - In multiple board systems, provide a separate extender board with a switch on each data line. Use this board to isolate the suspected board from the others. By closing one switch at a time and observing results, a specific bad bus can be detected.
  - With all data bus switches on the extender open and address bus switches closed, a failure can be localized without the rush of bad data feedback that can invalidate the control test stimulus patterns. Electronic switching will allow the ATE to control the testing and fault detection process.

- Use a current tracing tool to probe beyond the node level. Models are available that can be used off-line to trace the current on a faulty bus line to the source device.
- Do not use multilayer boards where there are bus lines coupling several devices. It is difficult to use a current tracer on such a board to discover which device is holding a bus line high or low.

RAMs, ROMs, Serial Registers

- Pattern sensitivity studies must be made for all large scale memory devices. Some current memories use as little as 28% of the real estate for the actual memory area. Circuit features other than the memory array are now primarily responsible for pattern sensitivity.
- Isolate large memory devices (1024 bits or larger) from significant amounts of standard logic. Consider the requirement of special test techniques and equipment for testing large memory.
- Provide access to control (enable) lines and output lines of memory devices at I/O pins or test points. If this is not possible, the memory devices should be mounted in sockets so that they can be removed. This permits a simpler test program for the PCB and a separate test for the memory device.
- Provide means in hardware or software to isolate ROMs from each other and from other bus elements. This can include providing ROM sockets to permit physical removal. This will allow isolation of address decode faults that result in incorrect data being read onto the bus.
- Provide a known PROM output state for every input address combination.
- Define any unused or don't care ROM states such that accidental entry into these states will bring the system back to a defined state.
- Provide means to disable, isolate, or remove each ROM individually in order to isolate the ROMs from each other and from other bus elements. The means can use hardware or software techniques, or a combination.

TESTABILITY TASK COMPENDIUM
Task Reference Number V2B

PHASE: VALIDATION

FUNCTION: System Design, Testability Incorporation

TASK TITLE: Provide for design compatibility of units to be tested with selected or available ETE or ATE.

TASK OBJECTIVES:

1. Discover and resolve any test interface incompatibilities between the design and the ETE or ATE.
2. Reduce or eliminate the need for a large number of unique Interface Device (ID) designs, and identify and specify the requirements for any unique ETE, ATE or IDs.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering	● System and UUT Performance Data	● $\overline{T}$ Design Guidelines
● Design Engineering	● UUT Interface Data	● ID Related Requirements, Interface Incompatibility Resolutions
● Logistics Engineering	● ETE/ATE Alternatives	● Interface Incompatibility Resolutions
● Test Equipment Engineering	● ETE/ATE Interface Data	● Interface Incompatibility Resolutions

COST TRADEOFF INTER-RELATIONSHIPS: Costs of development, acquisition and support are in general directly proportional to the diversity of test equipments and devices introduced with the prime system. Therefore new test equipments and the range and depth of interface devices and software should be kept to lowest possible minima.

TASK SYNOPSIS:

1. Task Requirements.

1.1 ETE/ATE Compatibility Verification Process(4)

- ATE/UUT* system compatibility can be determined by consideration of the following areas:

  a. UUT Packaging

  b. Physical interface between UUT and proposed ETE or ATE

  c. Electronic and power interface between UUT and proposed ETE or ATE

1.2 Specific Characteristics

- The following specific characteristics and features should be analyzed:

  a. Functional packaging scheme that results in UUTs that can be tested independently and singly

  b. Test point access and pin placement to permit testing to lowest level required and/or necessary for validity

  c. System and UUT design amenable to test by proposed ETE or ATE test stimuli and measurement devices. This will include accuracies required and available, accuracy ratio calculations, resource matching, and timing requirements.

  d. Uniform and compatible interface plugs and receptacles

  e. Power source compatibility

  f. Provision (to extent necessary) for manual intervention to allow operator to make adjustments to variable components

  g. Minimized need for equipment external to ETE or ATE to generate signals or monitor responses

2. Implementation.

2.1 In all interfaces, the testability engineer should be prompt, active and assertive. Testability is most effectively instilled into a system if design guidelines are provided when design commences. It is inefficient for the testability (or any other -ility) engineer to take the role of a post-design critic and difficult as well as uneconomical for designers to incorporate changes to documented designs when those changes reflect design criteria which could have and should have been disclosed during or ahead of design formulation. Timeliness is therefore a principal responsibility for the testability engineer.

---

* The term Unit Under Test (UUT) is used in a general sense and may be construed equivalently to LRU, SRU, WRA, SRA, Module or any accepted term for a testable component.

2.2 The following checklists provide guidance for the compatibility task.(4) The ETE and ATE Compatibility Checklist includes a numerical scoring feature for use in conjunction with the Completion Criteria of the Task Synopsis.

The checklists treat hierarchial indenture levels of prime systems. For clarity, two levels are illustrated as successively lower than the "LRU" level for which test initiation (in the context of this synopsis) is assumed. Definitions for LRU, SRU and sub-SRU as follows apply to terms used in the checklist.

LRU is a generic term that may be defined in terms of both Avionic equipment and Ground Systems equipment.

- For Avionic equipment or systems, it is Line Replaceable Unit.

Ground Electronics equipment it is Lowest Replaceable Unit.

LRU is defined as a unit which is designated by the plan for maintenance to be removed upon failure from a larger entity (equipment, system) in the latter's operational environment. (MIL-STD-1309-B)

A LRU is composed entirely of SRU's. A SRU is defined as follows:

SRU (Shop Replaceable Unit) - a generic term which includes all the packages within a LRU, which may include circuit boards, chassis, wiring harnesses and piece parts removed at the shop (intermediate or depot) level.

Sub-SRU - a generic term referring to a smaller circuit board or other device comprised of two or more piece parts mounted on a SRU.

ETE and ATE Compatibility Checklist - Candidate Quantitative Evaluation

Functional Modularity.

Determine if the LRU is functionally modularized at all levels of assembly/disassembly.

a. Determine:	Score
(1) Each LRU function is contained within a single SRU and each SRU function is contained within a Sub-SRU.	4
(2) The LRU is functionally modularized, but some SRU functions are not modularized within Sub-SRUs.	3

	Score
(3) A few LRU functions are contained on more than one SRU and/or most SRUs are not functionally modularized.	2
(4) Most LRU functions encompass more than one SRU.	0

Functional Independence. Determine if the LRU and its SRUs are capable of being tested without stimulation by another LRU or SRU and without simulation of another LRU or SRU.

a. Determine:	Score
(1) The LRU and all SRUs are functionally independent.	4
(2) Some SRUs require simulation within the ID using passive and/or simple active elements.	2
(3) Stimulation by another LRU or SRU is required, or complex simulation is required.	0

Adjustments. Determine if adjustments (e.g., trimming, tuning, alignment) must be made while testing on ETE/ATE. An adjustment includes any action that changes variable components such as potentiometers, variable capacitors, inductors, transformers, etc., that affect operation of the equipment.

a. Determine:	Score
(1) No adjustments or realignments are necessary for the LRU and its SRUs.	4
(2) A small number of simple non-interactive adjustments are required, but no complex adjustment or realignment is required.	3
(3) One or two SRUs require complex adjustment or realignment.	2
(4) The LRU or more than two SRUs require complex adjustment or alignment.	0

External Test Equipment. Determine whether external equipment is required to generate a stimulus or to monitor response signals.

a.	Determine:	Score
(1)	All stimulus generation and response monitoring can be accomplished by the target ETE/ATE.	4
(2)	Signal generation, synchronization, or waveshaping circuits are required within the ID.	2
(3)	Additional external test equipment is required.	0

Environmental. Determine if the LRU or the SRUs require special environmental considerations during test on the ETE/ATE, such as vacuum chambers, oil baths, shake tables, ovens, cooling air, and screen rooms.

a.	Determine:	Score
(1)	No special environment is required.	4
(2)	Forced air cooling or an electromagnetically shielded enclosure is required.	2
(3)	Other special environment conditions are required.	0

Stimulus and Measurement Accuracies. Determine the stimulus and measurement accuracies required for high confidence test.

a.	Determine:	Score
(1)	All tests can be performed on ETE/ATE at high confidence levels; i.e., stimulus is adequate and measurement is at least ten times more accurate than the tolerance on the UUT.	4
(2)	Measurement is at least three but less than ten times as accurate than the UUT tolerance.	3
(3)	Measurement is between one and three times more accurate than the UUT tolerance.	1
(4)	Stimulus and/or measurement accuracy is inadequate.	0

Test Point Adequacy. Determine if sufficient test points are provided for non-ambiguous fault isolation and for monitoring redundant circuits and BIT circuits.

a. Determine:

	Score
(1) Redundant and BIT circuits can be fully tested and test points at the output of each functional circuit permit direct non-ambiguous fault isolation.	4
(2) Indirect (non-signal tracing) troubleshooting and/or ambiguous fault isolation (within permissible limits per AR-10) is necessary.	3
(3) Redundant and BIT circuits not tested or there is excessive ambiguity.	0

Test Point Characteristics. Determine test point impedance and voltage levels.

a. Determine:

	Score
(1) Voltage is less than 350 VRMS and impedance is compatible with the ETE/ATE interface. LRU test points will drive up to ten feet of properly terminated coaxial cable.	4
(2) Voltage dividers and/or passive and simple active impedance transformation are required within the ID.	2
(3) Waveshaping and/or signal transformation is required.	2

Test Point Isolation. Determine if damage to a UUT will result from a short circuit between any test point and ground, or if wideband noise impressed on the test point will degrade performance.

a. Determine:

	Score
(1) Test points are insensitive to external disturbance and no damage results from a short circuit.	4
(2) Test points are sensitive to external disturbance but no damage results from short circuit.	2
(3) A test point short circuit will damage the UUT.	0

Power and Load Requirements. Determine the current and voltage required to power the LRU and the loads required to absorb the output power of the LRU.

a. Determine:	Score
(1) The power and load requirements can be met by standard ETE/ATE resources.	4
(2) The loads can be accommodated in a simple or intermediate ID.	3
(3) The quantity of loads is such that the ID is complex or ship's power or an external power source must be used.	0

Warm-up. Determine if the LRU and/or any SRU requires warm-up on ETE/ATE to ensure accurate test.

a. Determine:	Score
(1) No warm-up is required.	4
(2) Warm-up is less than 5 minutes.	2
(3) Less than 15 minutes.	1
(4) Greater than 15 minutes.	0

Evaluation: Generally, a UUT should score 31 or more total points (of 44 possible) to be considered acceptable in terms of testability. A score of zero in any of the eleven compatibility elements indicates a need for modification of the design.

Design Data Checklist - Qualitative Aids

General Interface Requirements

a. Unit-under-test orientation and/or environment should be non-critical.

b. All adjustment points should be clearly indicated, together with adjustment parameters.

c. There should be no EMI or RFI problems in testing the unit.

Electrical Interface and Parameters

a. Tolerances or limits of parameters are compatible with field preferably to factory requirements.

b. All special loading requirements are defined.

c. Special requirements are known for settling time relative to making a measurement.

d. Primary power requirements are clearly indicated, including maximum allowable variations in voltage and frequency.

e. Sequence of application or removal of power is identified necessary.

f. Each input and output signal is completely defined.

g. Each test point signal is completely defined.

h. High frequency line-lengths are non-critical.

i. Trigger or synchronizing inputs are provided from ETE or ATE.

3. Completion Criteria.

The task may be considered complete when results are incorporated in the Testability Analysis Report, Task V5A. Further iteration and refinement will occur in the Full Scale Development Phase, Function F1.

TESTABILITY TASK COMPENDIUM

Task Reference Number V2C

PHASE: VALIDATION

FUNCTION: System Design, Testability Incorporation

TASK TITLE: Provide direct designs and test sequences using parts selected for testability

TASK OBJECTIVE: Assure that the system design uses parts, straightforward equipment designs, and software that have proven testability characteristics in every possible instance.

DESIGN DISCIPLINE TESTABILITY ($\bar{T}$) INTER-RELATIONSHIPS:

$\bar{T}$ INTERFACE	INPUT TO $\bar{T}$	OUTPUT FROM $\bar{T}$
● System Engineering		● Testability Design Guidelines
● Design Engineering	● Logic diagrams, Parts Lists	● Testability Analysis Results
● Application Software	● Test Program Listing	● Critique Listing for Comments, Traps

COST TRADEOFF INTER-RELATIONSHIPS: Favorable parts selections and design implementations serve to reduce life cycle costs of testing and test support. Also, use of common approved parts as feasible reduces acquisition as well as support costs.

TASK SYNOPSIS:

1. Task Requirements.

1.1 The following should act as a constraint against the design.

a. Parts Selection.

In selecting between parts, each with satisfactory performance characteristics, give preference to integrated circuit components and assembled modules which have proven satisfactory testability characteristics, and to those integrated circuits for which sufficient disclosure of internal IC structure and failure modes have been provided as a basis for effective economical testing.(5)

b. Design Simplicity.

If permitted by performance requirements, provide structured, straightforward designs using standard components rather than random designs using nonstandard components. In generating test sequences, a regular, systematic test is preferable to a test which employs subtle tricks for length minimization.(5)

2. Implementation.

2.1 Parts Design and Software Guidelines.

The following guidelines condensed from the literature-at-large of testability philosophy, should be used to incorporate parts, equipment designs, and software of proven testability characteristics into the prime system design.

a. Minimum functional, fault detection and fault isolation test costs should be considered a prime requirement during the design stage.

b. All constraints imposed upon the board assembly testing should be covered in the initial design plan and not as an afterthought.

c. Provide the results of testability studies to the designer in addition to initial guidelines.

d. The evaluation of hardware chosen for developmental testing should include design and testability constraints.

e. Where possible, use standard highly testable components and consistent part orientation.

f. Provide simple and short reset sequences which the tester can control (See Initialization-Validation phase, Task V2E).

g. Provide tester access to the main buses.

h. Limit the number of logic families.

i. Make testability requirements known to mechanical, electrical, and software design.

2.2 General Good Engineering Practices.(6)

a. Electrical Design Ideas

(1) Make system independent of supply-voltage sequences.
(2) Allow shorting of input-output lines.
(3) Allow for reasonable capacitive loads for both input-output lines and test points.
(4) Allow shorting of adjacent connector pins.
(5) Try to prevent "Domino" failures.
(6) Use only one logic family.
(7) Allow internal clocks to be easily disabled.
(8) Allow for external initialization.

b. Mechanical Design Practices

(1) Use zero insertion connectors.
(2) Use a defeatable keying system for using extender cards.
(3) Leave space between components.
(4) Use consistent part orientation and standard parts.
(5) Limit the number of logic families within each assembly.
(6) Use functional packaging.

c. Logical Design Practices

(1) Use built-in test (BIT).
(2) Use selected test and control points.
(3) Avoid wired ANDs and ORs in system.
(4) Use wired AND for test purposes.
(5) Interrupt feedback and redundant loops.
(6) Break up long counter chains.

d. Managerial Practices

(1) Allow the circuit designer to be responsible for tests and test programs.
(2) Use configuration control.
(3) Establish achievable goals.
(4) Make testability requirements a peer set with the disciplines of reliability, maintainability, et al.

2.3 Quality Criteria.

When considering only the inherent testable design aspects of testability, several measures may be used which give a gross indication of potential testability and point to problem areas. Most of these simply make use of a checklist (or perhaps a weighted checklist) for the presence or absence of testable characteristics. For example: (7)

a. Is the design straightforward and regular?
b. Is the circuit electrically and mechanically compatible with the available ETE/ATE?
c. Are conservative timing and signal tolerances used?

d. Is the design partitioned by function?

e. Does the circuit include a master reset?

f. Are control and data paths separated?

g. Are critical nodes brought out to test points?

h. Are non-standard components used?

i. Are the test sequences regular and systematic?

These and similar elements should be adapted, tailored and applied as criteria during working and formal design reviews.

3. Completion Criteria.

The task is completed for a specific program phase when the specifications and/or design documentation for that phase are found to meet the design criteria and are accepted by the Government.

TESTABILITY TASK COMPENDIUM
Task Reference Number V2D

PHASE: VALIDATION

FUNCTION: System Design, Testability Incorporation

TASK TITLE: Design units with testable physical partitioning

TASK OBJECTIVE: Assure that the systematic partitioning at each successive indenture level of the prime system provides for ease of functional testing and fault isolation as well as for ease of repair by removal and replacement.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering	● System Weight & Size, System Partitioning	● Testability Requirements, Design Guidelines
● Design Engineering	● Subsystem Partitioning	● Testability Requirements, Design Guidelines
● Maintainability Engineering	● Maintainability and Repairability Requirements	● Merge Maintainability/ Testability Requirements
● Life Cycle Costing	● Cost Analysis, Tradeoffs	● Critique of LCC Tradeoffs

COST TRADEOFF INTER-RELATIONSHIPS: Reduced costs result from proper partitioning, with benefits beginning to manifest themselves during developmental test. Acquisition cost reductions occur in the areas of spares and test program sets. Operating and maintenance skill levels and manhour expenditures will also be reduced.

TASK SYNOPSIS:

1. Task Requirements.

1.1 The following should guide the design:(4)

a. The physical partitioning of an equipment into modules will be based, in part, upon the enhancement of the fault isolation process.

b. The maximum number of UUT pins will be consistent with the interface capabilities of the proposed ETE or ATE.

c. Modules will be designed to be easily removable plug-in units and, whenever practical and economically justified, will be of such cost and reliability that they may be considered for discard-on-failure.

d. Any individual module should be limited to one circuit technology, containing only analog or only digital circuitry, whenever practical.

e. Where practical, circuits belonging to an ambiguity group will be placed in the same package (component, module).

2. Implementation.

2.1 Physical partitioning is that step in the design process which provides a system that is physically and functionally packaged to facilitate testability and maintainability. This step normally occurs after completion of system logic design. It is accomplished by breaking the complete system into smaller subsystems which are then packaged individually.

Constraints include a limit on the size and number of components in a subsystem as well as on the number of interconnections between subsystems.

2.2 Appendix A(7) is a (condensed) heuristic method for enhancing testability during the partitioning of a large digital system. The test flow model can be used for test point insertion and can easily be incorporated into other partitioning procedures which optimize primary constraints such as the number of elements or the number of I/O pins.

2.3 One objective of physical partitioning is optimized system/black box repairability.(8) Three important avionics equipment repair problem areas to which this concept is applicable are: excessive subsystem packaging complexity, excessive black box weight, and black box troubleshooting/repair.

Fewer black boxes for minimum system packaging complexity benefits primarily the organizational maintenance level with major reductions in organizational level support requirements in the areas of labor, training (especially OJT), technical data, and supply.

To avoid excessively heavy black boxes, partitioning units into two or more units can result in a weight such that handling equipment would not be required and manhandling (the preferred alternative) would be facilitated. Partitioning subsystems into packages which are easily handled by two-man teams would benefit all maintenance levels. Special handling equipment, as presently authorized for organizational level handling of heavy units, would not be required in the future, thereby significantly reducing support equipment requirements,

Optimized partitioning, in conjunction with new interface concepts and integrated electronics, can be applied to the design process to simplify troubleshooting and repair in the intermediate shop. Complex units such as presently employed in the weapons control and weapons delivery equipment, are difficult to troubleshoot to the module level. This problem is a result of various factors including number of modules per black box, chassis complexity, and test equipment limitations. Partitioning into fewer modules, and in some cases eliminating the chassis and using cable assemblies for module interconnections (example is the AN/ARC-164 radio design), can greatly simplify unit troubleshooting. Further partitioning of modules into smaller plug-in throwaway sub-modules (as used in the Mark V receiver/transmitter) is an additional refinement of the partitioning concept for further simplification of the repair process. Such partitioning concepts would simplify troubleshooting/repair and would have significant impact on the intermediate and depot levels of maintenance, primarily in the area of labor.

2.4 Quality Criteria.

The following relevant parameters may be adapted quantitatively or qualitatively to formulate criteria for measuring the effectiveness of the partitioning task as tailored to specific programs and systems.

PARTITIONING RELEVANT PARAMETERS	POSITIVE EFFECT
● Element* Quantities	● Simplified Troubleshooting/Repair Procedures ● Minimized Organizational Level Test Support Requirements
● Element* Weights	● Obviated Special Handling Equipment
● Element* Volumes	● Easier Handling
● Element* Complexities	● Simplified Troubleshooting/Repair Procedures
● Number of Interconnections (I/O Pins)	● Less Complex Interface, Improved Testability

3. Completion Criteria.

The task must be considered complete when the development specifications of Task V6 have been approved by the Government.

* "Element" may be construed as a UUT at any level of hardware hierarchy.

APPENDIX V2D1 (7)

MODELING THE SYSTEM

Assume that the system is described as a set of interconnected elements, each having its own input and output pins. In addition, the system itself will have a set of primary inputs and outputs. All interconnections within the system will be one of three types: (1) a primary input to an element input, (2) an element output to a primary output, or (3) an element output to an element input. (See Figure 1.) Assume that all testing of the system must be accomplished through the primary inputs and outputs.

The functional nature of the elements will not be considered; they may range from simple gates up to entire cabinets. Each element will be assigned a measure, t*, which reflects its testability relative to the other elements in the system. The nature of this "measure of testability" is left to the user. It could be as sophisticated as the number of tests required to detect some fixed percentage of all stuck-at faults in that element or as simple as a mere component count, as long as it accurately reflects the relative test requirements of the elements. Figure 1 shows a set of such measures assigned to the elements of the system with a "test load" of 1300.

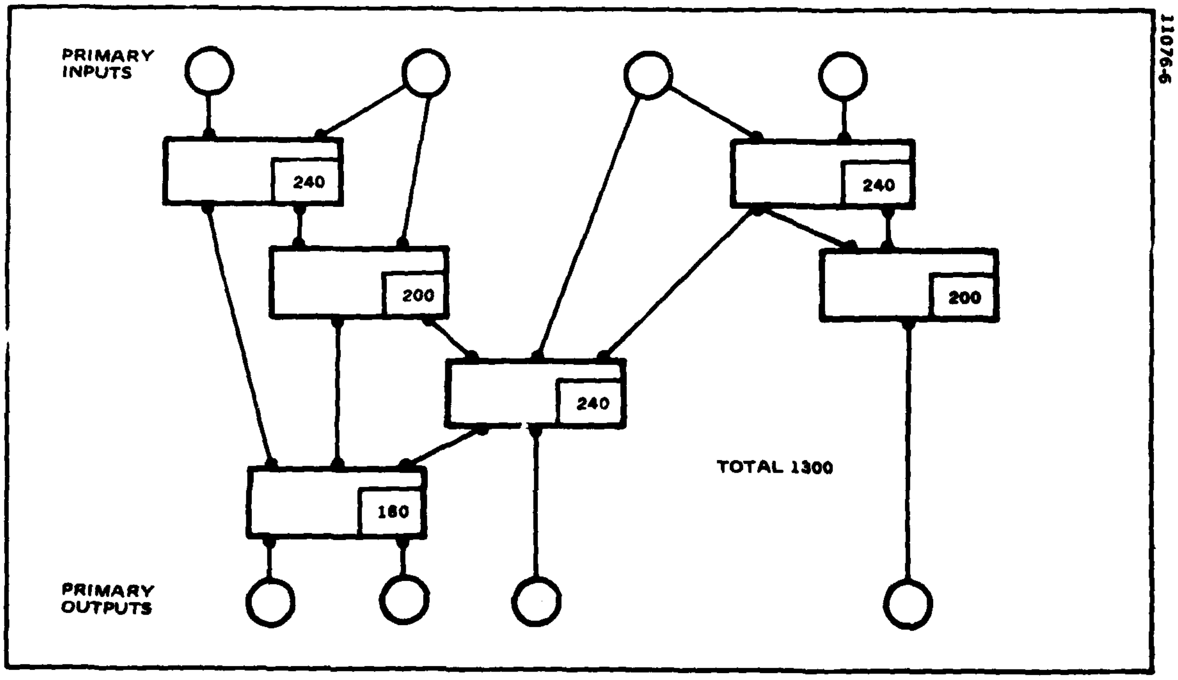


Figure 1. Model of a System

Secondly, consider the accessibility of these elements as viewed from the primary inputs and outputs. An element with a relatively small testability measure may be quite difficult to test simply because it is buried deep within the system.

Picture the testing of an element (with measure t*) as a flow process: a set of t* input tests or excitations are introduced onto the primary inputs of the system. These excitations propagate through the system to the inputs of the element involved. They pass through the element and emerge on the outputs as a set of t* detections. These detections then continue on to the primary outputs of the system where they can be examined for indications of possible malfunctions.

The nature of the flow will change (from excitations to detections) when it passes through the element.

We make one further assumption that no feedback paths be present. The partitioning constraints prohibit the cutting of some or all of such feedback paths. Now all elements appearing in these paths can be merged into a single element whose testability measure is then adjusted accordingly. (Straightforward techniques for accomplishing this are described by Ramamoorthy & Chang*.) If feedback paths still exist, various interconnections may be temporarily deleted.**

First distribute the t* detections coming out of the element forward to the primary outputs using the following "divide equally" assumptions:

(1) The t* detections are assumed to divide equally among the output pins of the given element.

(2) If an output pin drives a primary output, all of its flow will be assumed to go to that output. Otherwise, it will be assumed to divide equally among the input pins which it drives.

(3) The total flow into any other element will be assumed to divide equally among that element's output pins.

Figure 2 shows the application of these rules to the 200 unit flow out of element A. The flow divides into 100 on each of the two output pins. These flows then continue to the inputs of elements B and C. In element B, the 100 unit flow again divides equally with 50 units going on to a primary output and 50 units to element C. Finally, the total flow of 150 into C divides equally between its two output pins and continues to the respective primary outputs.

---

* C.V. Ramamoorthy & L.C. Chang, "System Segmentation for the Parallel Diagnosis of Computers", IEEE Trans. on Computers, Vol. CT-20, pp 261-270, March 1971.

** U. R. Kodres, "Partitioning and Card Selection" in Design Automation of Digital Systems (M. Breuer, ed.) Englewood Cliffs, N.J.: Prentice-Hall, 1972.

The set of "divide equally" assumptions may be used to propagate the 200 excitations of element A backwards to the primary inputs. Figure 3 shows the result of this step. (Detections have been underlined to distinguish them from excitations.)

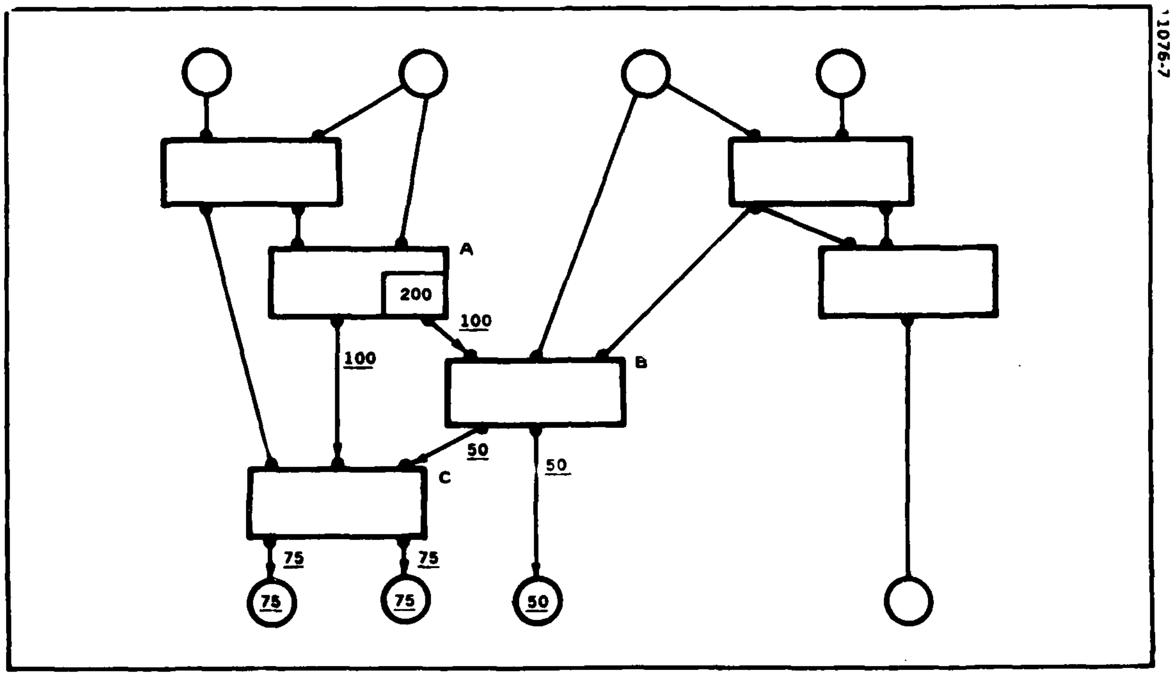


Figure 2. Distributing Detections

Now repeat this flow generation process for each of the elements within the system. Then add up the resulting flows to obtain the completed test flow model shown in Figure 4. The flow on each lead is designated by two numbers denoting the number of detections (underlined) and the number of excitations which that lead will carry.

THE TEST FLOW MODEL

Note that for any element with measure t*:

$$d_{out} = d_{in} + t^* \qquad (1)$$

and,

$$e_{out} = e_{in} - t^* \qquad (2)$$

where d and e denote detections and excitations.

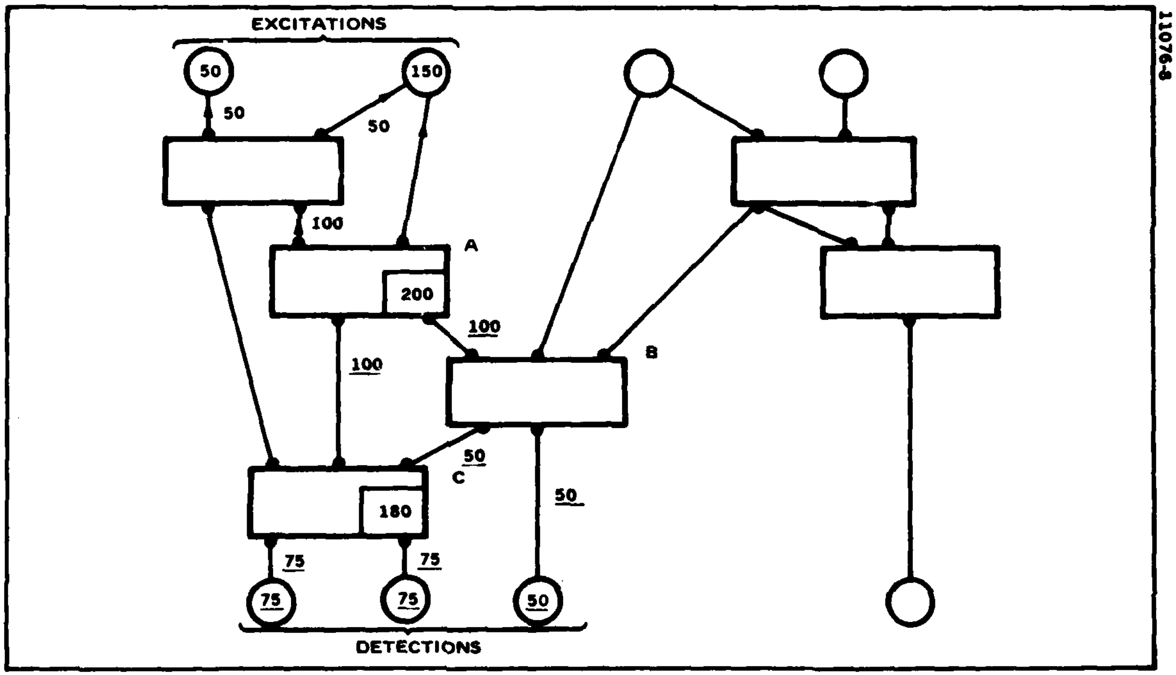


Figure 3. Distributing Excitations

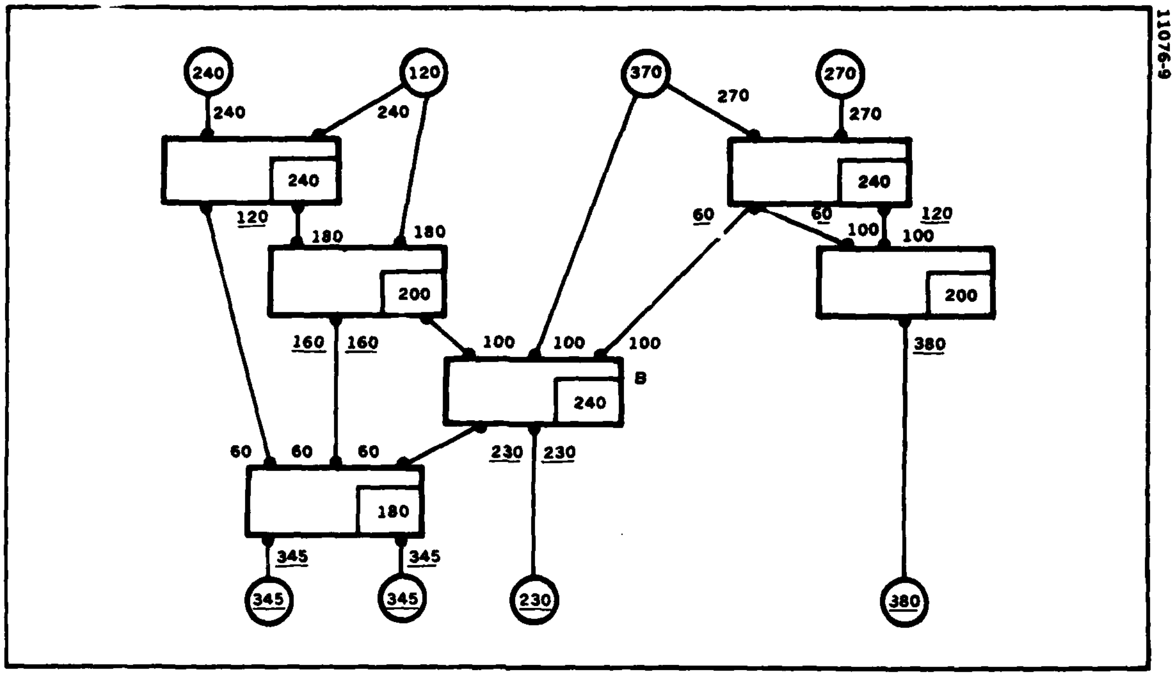


Figure 4. The Test Flow Model

For example, the flow into element B (Figure 4) is 520, 220 detections and 300 excitations. The total out is also 520 composed of 460 detections and 60 excitations - the result of 240 excitations changed to detections.

If we partition the elements of the system into two disjoint sets, then the set of leads interconnecting these two sets comprise a cut. See (*) for a precise definition. The flow across a cut is equal to the total flow, T, through the entire system. For any cut, C, if we add up the detections and excitations on its individual leads, then

$$\sum_{C} d_i + \sum_{C} e_i = T \qquad (3)$$

Consider the 6 lead cut in Figure 5. The sum of the detections is 120+160+160+0+60+380 = 880. The sum of the excitations is 60+60+100+100+100+0 = 420.

$$T = 880 + 420 = 1300$$

(*) C.Berge, The Theory of Graphs, New York: John Wiley & Sons, 1962

The detection flow of 880 equals the sum of the measures of the four elements above the cut and the excitation flow of 420 corresponds to the sum of the measures of the two elements below the cut.

The individual detection and excitation flows tell, respectively, exactly how this load is divided above and below the cut. This properly can be especially useful during the partitioning process.

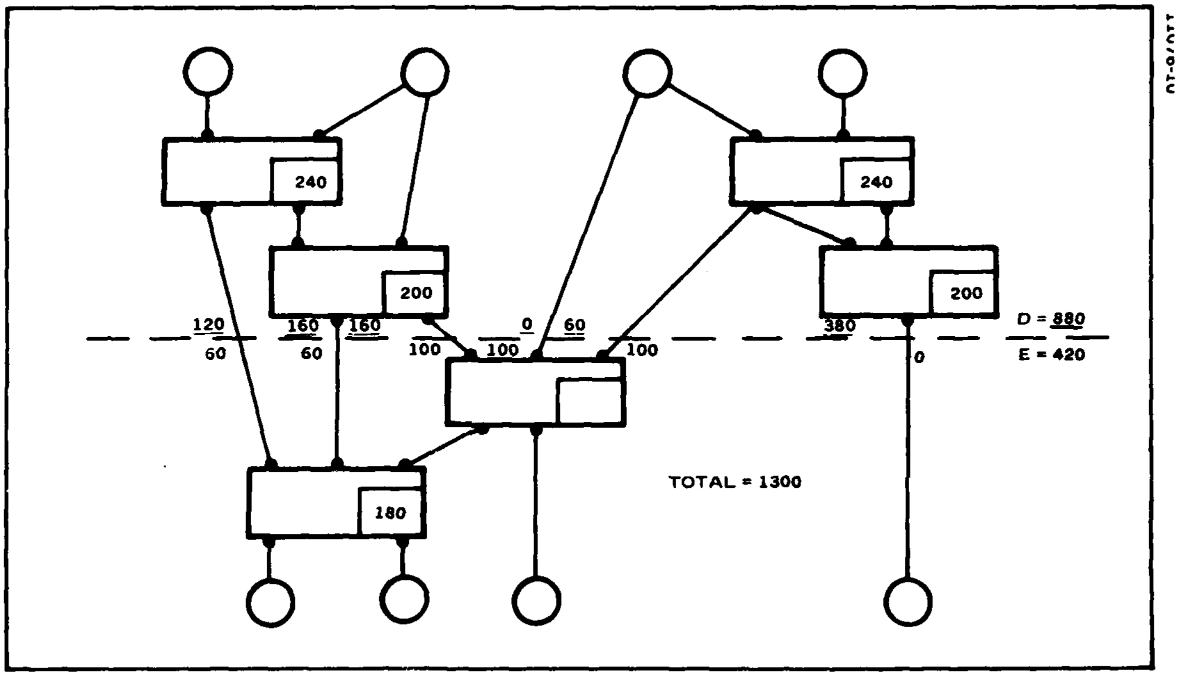


Figure 5. A Cut

We wish to partition the system in Figure 4 into two subsystems so that the total test load is divided as nearly equal as possible between the two subsystems.

Assign each lead, i, a weight, w_i, which is equal to the absolute value of the difference between its detections and its excitations, i.e.,

$$w_i = (d_i - e_i) \qquad (4)$$

Figure 6 shows the lead weights which would result in this example. Note: leads near the center of the system tend to have small weights, those near the extremes have large weights. The value, c of any cut - i.e., the sum of the weights of the leads in the cut - gives any upper bound on the difference of T_1 and T_2 (the test loads of the resulting subsystems):

$$|T_1 - T_2| \leq c$$

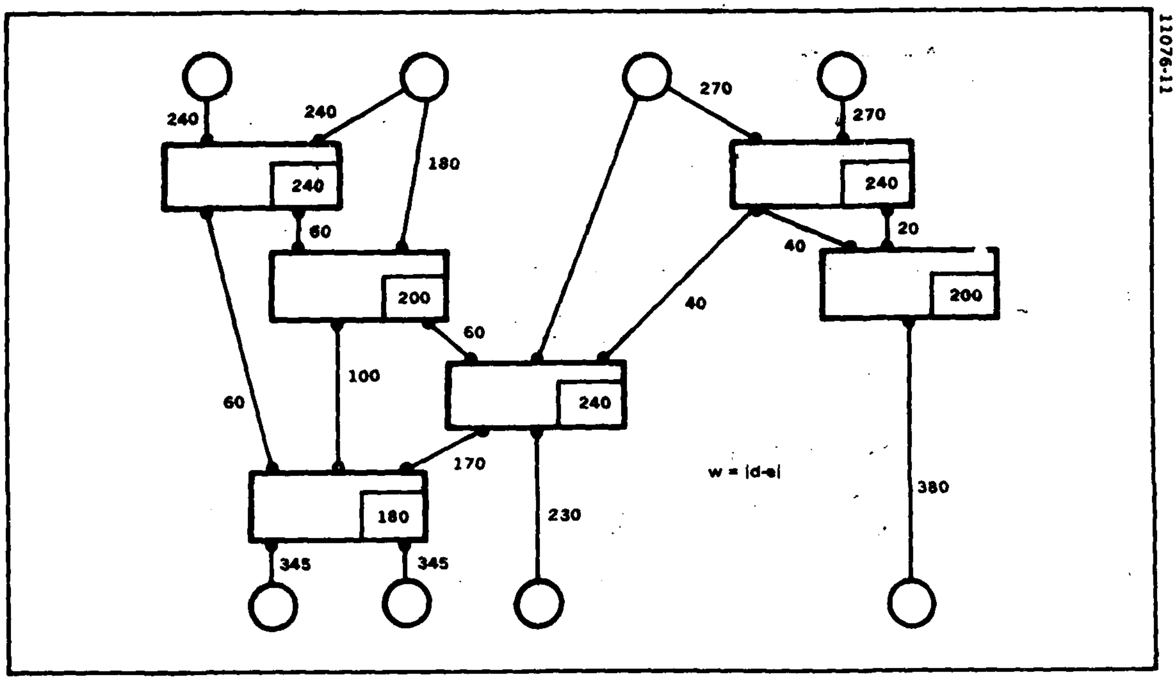


Figure 6. Weighted Leads

If we find the minimum cut of the system (with value c*), then the resulting test loads will differ by at most c*.

A basic algorithm from network flow theory (The Min Cut/Max Flow Algorithm) finds precisely this cut.

Figure 7 shows the minimum cut which results when this algorithm is applied to the example. The test load divides as:

$$T_1 = 680 \text{ and } T_2 = 620.$$

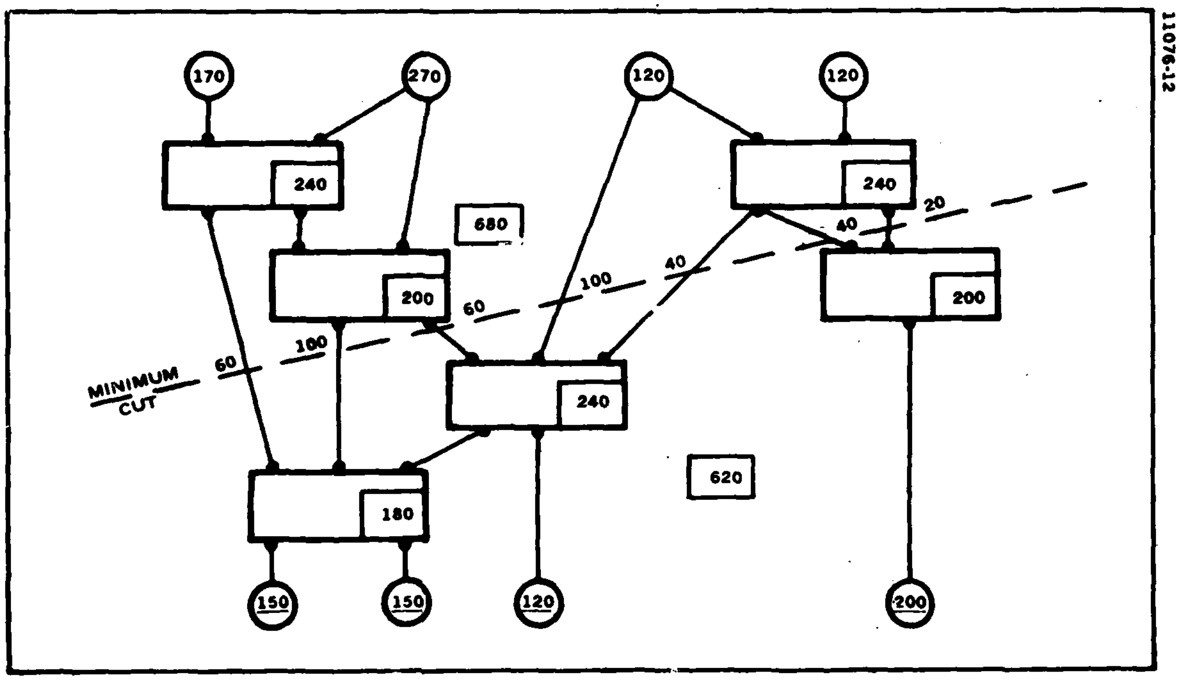


Figure 7. The Minimum Cut

This same technique (weighting the leads and then using the Min Cut/Max Flow algorithm) can be generalized to partitioning the system into p parts by simply adjusting the weighting procedure. Define each lead weight as:

$$w_i = \left|(p - 1)d_i - e_i\right|. \qquad (6)$$

The test load above the resulting minimum cut will be roughly T/p, where T is the total test load. Once the subsystem generated by this cut is removed, the test flow for the remainder of the system may be recalculated and the process repeated for p-1, etc. A computer program implementing this procedure has proved to be quite fast and effective. Its use is illustrated below.

Figure 6 shows the lead weights which would result in this example. Note: leads near the center of the system tend to have small weights, those near the extremes have large weights. The value, c of any cut - i.e., the sum of the weights of the leads in the cut - gives any upper bound on the difference of T_1 and T_2 (the test loads of the resulting subsystems):

$$\left| T_1 - T_2 \right| \leq c$$

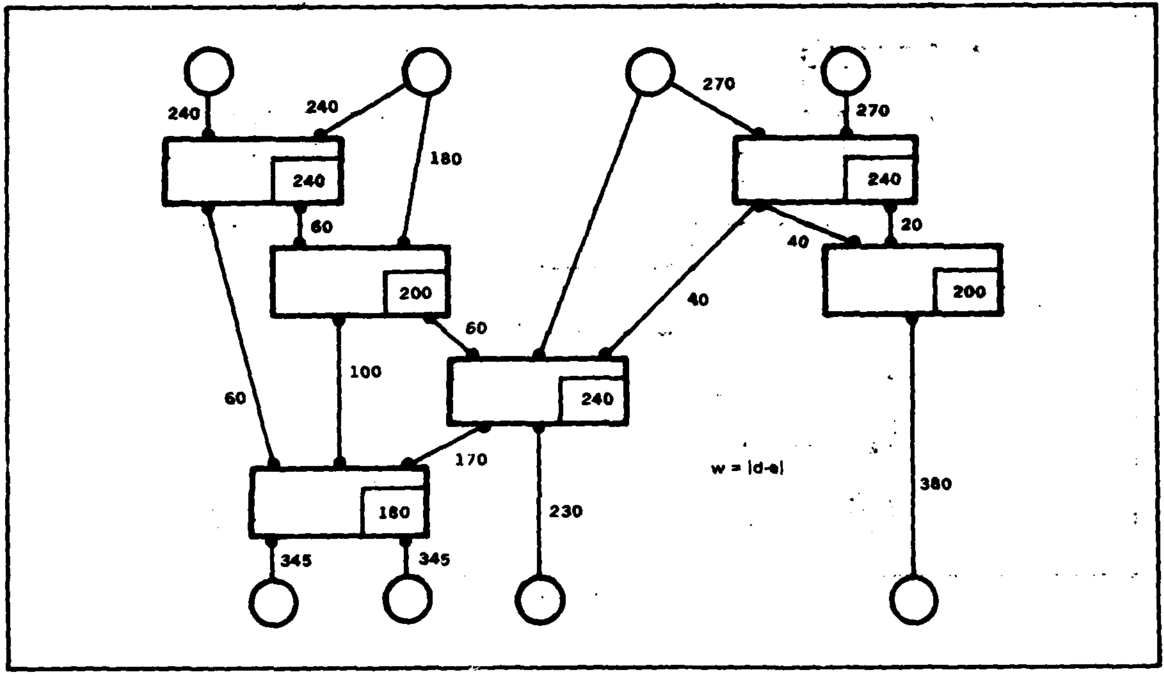


Figure 6. Weighted Leads

If we find the minimum cut of the system (with value c*), then the resulting test loads will differ by at most c*.

A basic algorithm from network flow theory (The Min Cut/Max Flow Algorithm) finds precisely this cut.

Figure 7 shows the minimum cut which results when this algorithm is applied to the example. The test load divides as:

$$T_1 = 680 \text{ and } T_2 = 620.$$

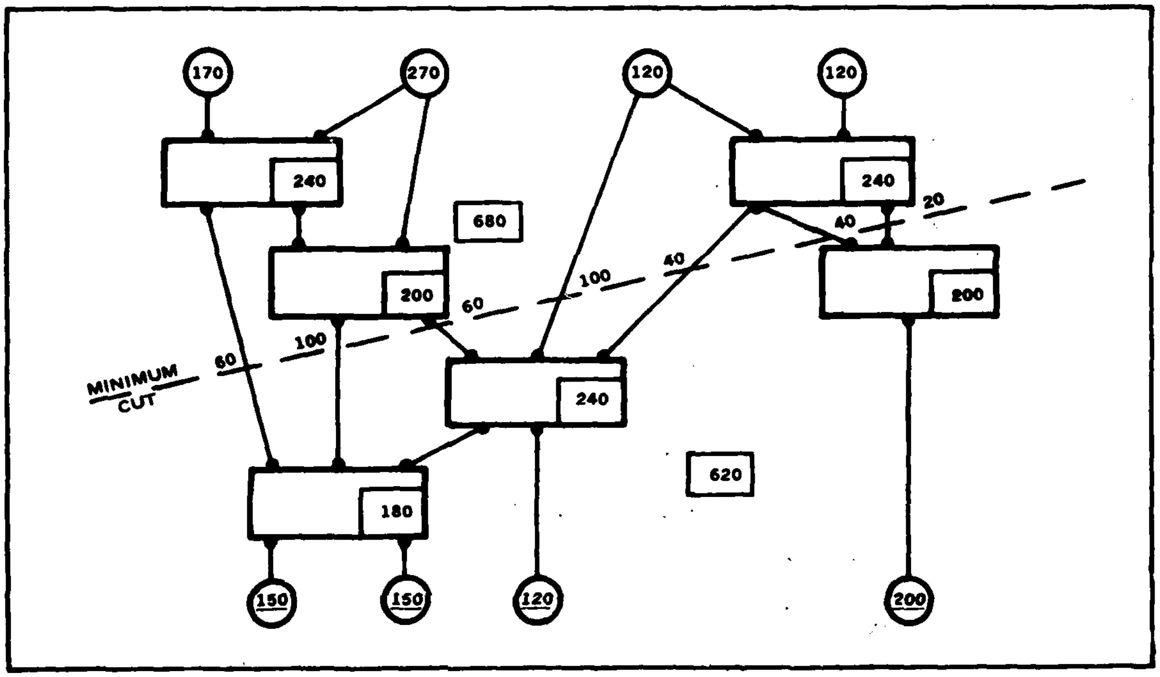


Figure 7. The Minimum Cut

This same technique (weighting the leads and then using the Min Cut/Max Flow algorithm) can be generalized to partitioning the system into p parts by simply adjusting the weighting procedure. Define each lead weight as:

$$w_i = \left|(p - 1)d_i - e_i\right|. \qquad (6)$$

The test load above the resulting minimum cut will be roughly T/p, where T is the total test load. Once the subsystem generated by this cut is removed, the test flow for the remainder of the system may be recalculated and the process repeated for p-1, etc. A computer program implementing this procedure has proved to be quite fast and effective. Its use is illustrated below.

Partitioning problems usually involve a number of considerations which must be examined. Two of the most common are the total size of the elements in each subsystem and the number of interconnections between subsystems. To incorporate size considerations into the Min Cut procedure initially distribute the numerical size, s*, of each element through the system just as we did for the t*s. Each lead i will then end up not only with a d_i and e_i, but also with an s_i and an s'_i. Across any cut $\sum s_i$ will equal the total size of the elements above the cut and $\sum s'_i$ the total below the cut. If a partition which has been chosen to optimize testability is found to violate a size constraint, then the lead weights may be changed to a weighted sum of the s's as well as the d's and e's. The exact nature of this sum will depend on both the desired size of the partition and on its criticality relative to testability considerations.

When the number of leads in an indicated cut is found to be excessive, a properly chosen weight, ℓ, can be added to the other lead weights so that the resulting minimum cut will tend to minimize the number of leads as well.

TEST POINT INSPECTION

On every primary input pin we have a numerical indication of its share of the input excitation load, and the detection load on the primary output pins.

Consider the test flow model of Figure 4, and assume that this represents a subsystem resulting from the partitioning process. If we are to insert a single test point where should this be done to best enhance the subsystem's testability?

Choose the internal lead that carries the largest number of detections. In this case, the internal lead from B to C (with 230) would be chosen and the test point inserted accordingly. Modify the model to reflect an insertion at this location. By following the same rules by which it was originally distributed and decrementing the flows accordingly, the resulting test flow is that shown in Figure 8.

Another possibility is to consider "cutting" an internal lead - i.e, to replace the lead by two test points, one a primary input and the other a primary output. During normal operation these points would be interconnected, but removed during the testing phase. A good candidate for such a cutting process is that internal lead with the largest total test flow. In Figure 4, we select the internal lead into element A having a total flow of 300. Figure 9 shows the resulting test flow.

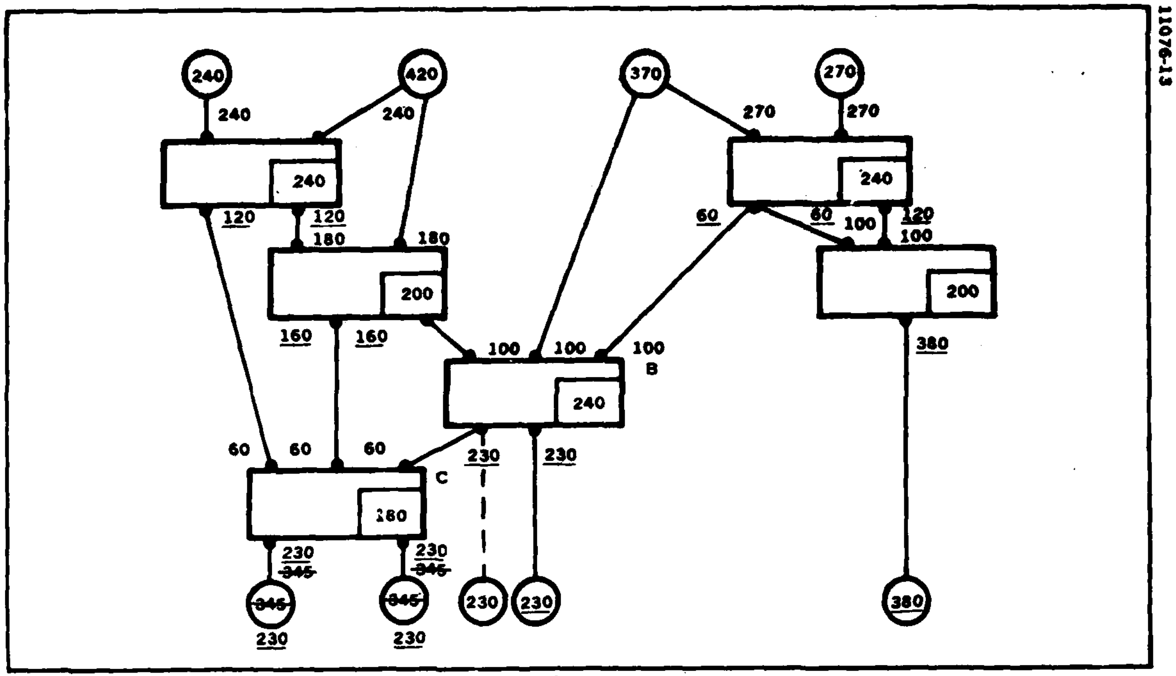


Figure 8. Adding a Test Point

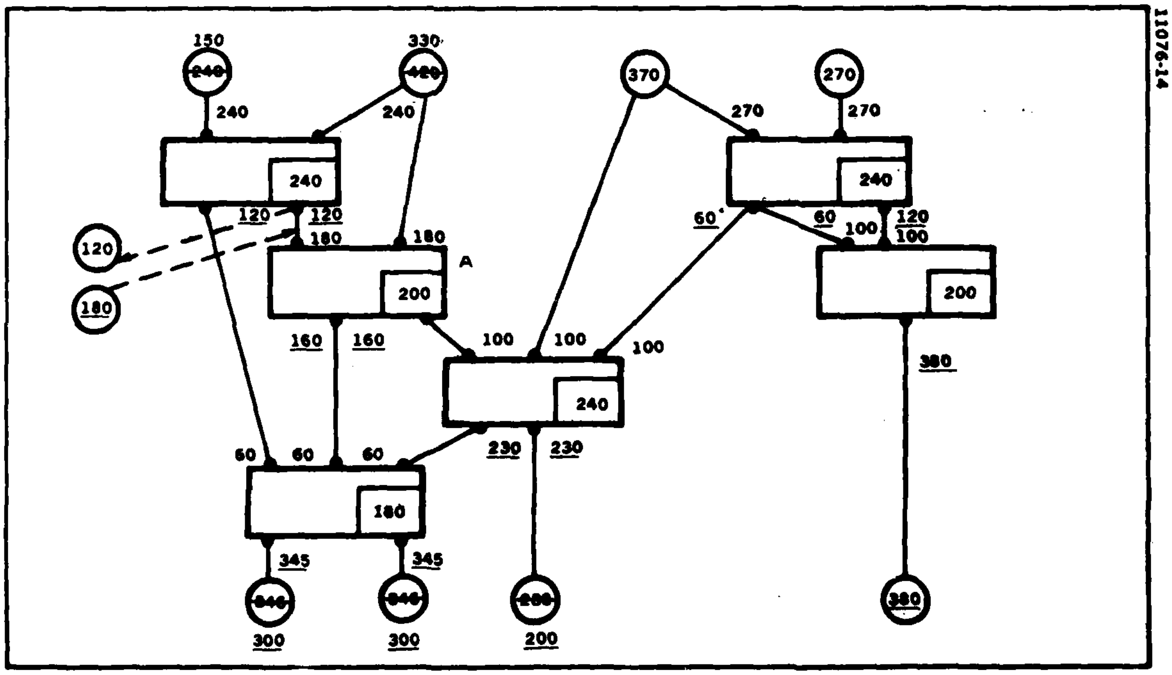


Figure 9. Cutting a Lead

TESTABILITY TASK COMPENDIUM

Task Reference Number V2E

PHASE: Validation

FUNCTION: System Design, Testability Incorporation

TASK TITLE: Incorporate initialization into design

TASK OBJECTIVE: Assure that the design provides circuitry, firmware and software or application programs to establish a well-defined unique initial or starting state at the beginning of, or at prescribed points in, a functional test or a fault isolation process.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering	● System level BIT data	● Initialization Requirements
● Design Engineering	● Number & placement of UUT test points and test control inputs	● Initialization Requirements
● Application Software	● BIT programs	
● Life Cycle Cost	● Cost guidance	● ATE software tradeoffs
● System Test	● Evaluation of demonstrability of initialization	● $\overline{T}$ demonstration candidate elements

COST TRADEOFF INTER-RELATIONSHIPS: Well-designed initialization capability reduces ETE/ATE and BIT software costs and reduces the costs associated with field maintenance testing.

TASK SYNOPSIS:

1. **Task Requirements.**

Initialization requirements pertain primarily to digital applications, although analogous requirements may well exist to establish a basic state for analog testing.

1.1 Two features which characterize initialization in the design are:

a. The system/equipment design should be such that it has a well-defined initial state to commence the fault isolation process. Non-achievement of the correct initial state should be annunciated to the operator along with sufficient signature data for fault isolation.

b. The system/equipment should be designed to initialize to a unique state such that it will respond in a consistent manner for multiple testing of a given failure. (5)

2. <u>Implementation</u>.

2.1 The following are suggested techniques for achieving initialization of the logic elements and are not all inclusive. (10) (11)

a. Use wired AND (OR) for external control connections (Figure 1) as a means of initializing the logic elements. This wired-or technique is safe for DTL circuits and for standard and low-power TTL circuits. But high-power and Schottky TTL circuits cannot be driven low safely for more than one second. Where an output of an IC feeds back into the IC (as in a flip-flop, shift register or counter), the output may be driven low using the wired-or technique. For example, the "slave" stage of a master-slave flip-flop can be initialized by driving the Q or $\overline{Q}$ output low, but the "master" stage remains unchanged. See Figure 2.

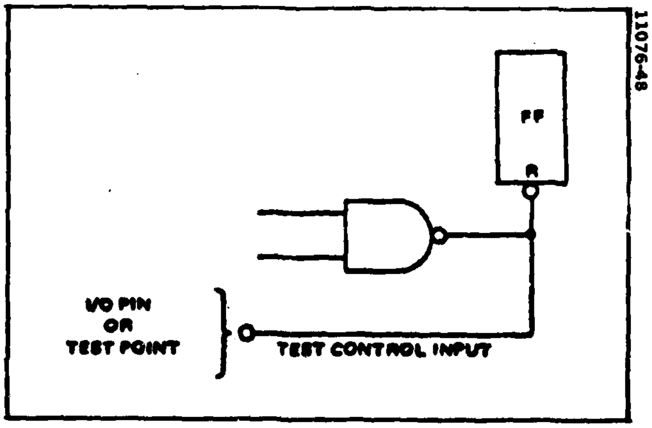


Figure 1. Wired OR for Test Control Input

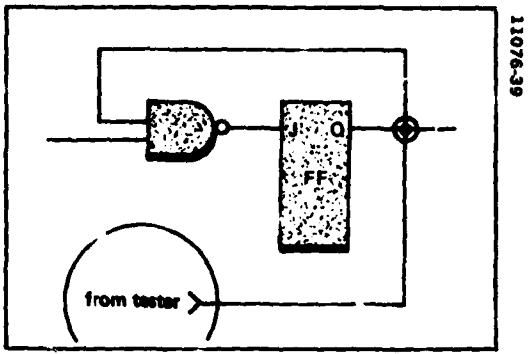


Figure 2. Wired OR Feedback Circuit

b. Initialize all sequential circuits to a known start condition utilizing the shortest possible sequence, ideally one transition, and never more than 20 transitions.

c. Provide means of initializing all sequential logic elements from I/O pins or test points. This applies to flip-flops, counters, registers, memories, etc. (Figure 3).

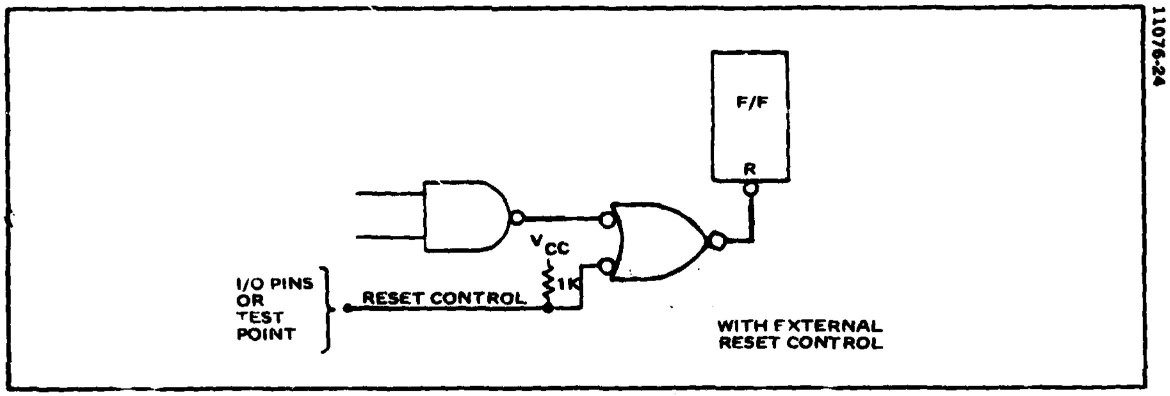


Figure 3. Flip-Flop Modified for External Reset Control

In the same sense, if an I/O pin or test point is not available, provide initialization by means of "power-up". (Figure 4).

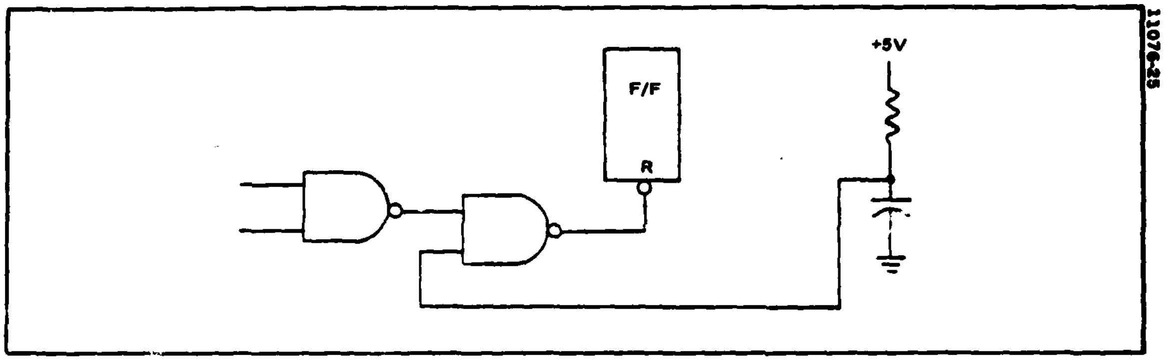


Figure 4. Power-Up Reset

d. The same pull-up resistor can be used for several sets or resets on different memory elements. (Figure 5).

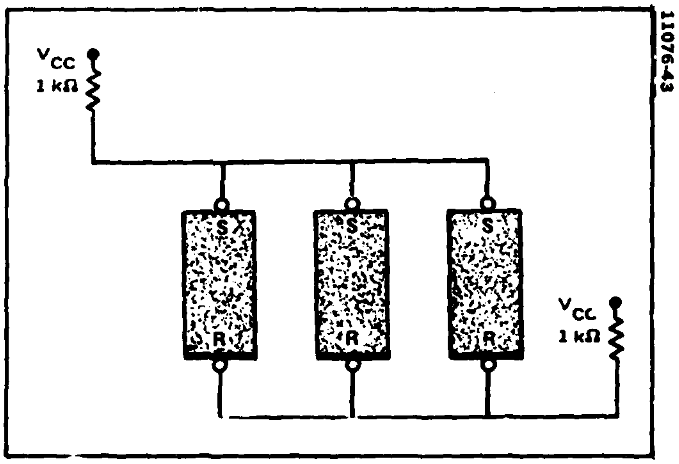


Figure 5. Pull-Up Resistor Multiple Reset

e. If a set-reset line is tied directly to V_{CC} or ground, it cannot be driven by the tester. Instead, a source of a logic high should come from a pull-up resistor and a source of a logic low should come from an inverter whose input is pulled high.

2.2 A verification-of design-review is necessary since initialization cannot be fully validated until the performance of a maintainability or testability demonstration. Internal low-level design reviews should therefore be conducted, either by supervision or by one designer reviewing another's work. The baseline verification guide is to verify that all memories, flip-flops and registers can be initialized to a known state.

Coordination should also be effected with the planning for the testability or maintainability demonstration for evaluation of the demonstrability of the initialization characteristics of the system.

2.3 A procedure should be tailored to each design program to obtain certification as a result of internal design review processes and incorporation of initialization measurement into the T or M demonstration planning.

3. Completion Criteria.

The task is considered complete when the development specifications (Task V6) are approved by the government.

TESTABILITY TASK COMPENDIUM

Task Reference Number V2F

PHASE: VALIDATION

FUNCTION: System Design, Testability Incorporation

TASK TITLE: Incorporate controllability into design

TASK OBJECTIVE: Assure the design provides adequate test control over system/UUT operation.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● Design Engineering	● Logic Diagrams, Schematics	● Controllability Requirements & Guidelines
● Application Software	● Test Programs, Logic Flow Diagrams	● Controllability Requirements & Guidelines
● Life Cycle Cost	● LCC Information	● Test Cost Parametric Data

COST TRADEOFF INTER-RELATIONSHIPS: The degree of controllability built into the hardware and software design results in proportionally less complex test equipment and test programs with reduced acquisition costs therefor. Also, downstream operations and maintenance costs associated with reduced test and repair times and corresponding maintenance labor and spares reductions provide further LCC savings.

TASK SYNOPSIS:

1. **Task Requirements.**

1.1 Controllability is defined as "an attribute of equipment design which defines or describes the degree of test control which may be exercised at internal nodes of interest."(1)

This task guides the designer in developing circuitry such that BIT and/or BITE and/or ETE simplifies testing via controls over internal module and component operation. The requirement is for valid detection and isolation of faults and failures.

2. **Implementation.**

Controllability of test is accomplished essentially at internal nodes;

therefore, the implementation techniques provide for access where necessary for additional data paths and circuitry. The literature is fairly extensive in coverage of techniques applicable to digital circuitry, while analog test is covered by inference or by references to textbooks.

2.1 The following list of controllability design guidelines apply to digital equipment.(12)

a. A digital circuit design should avoid large memory devices such as 1024 bit or larger RAMs plus significant amounts of standard logic, since such devices require special test techniques and equipment.

b. Dynamic devices which require periodic refresh, e.g., dynamic memories and some microprocessors, require special tester circuitry and should be avoided. Provision of a separate clock for these devices improves testability.

c. Complex LSI circuits (microprocessors, I/O controllers, etc.) should be installed in sockets because existing logic testers are not satisfactory for testing boards with such devices. If sockets are prohibited, provide self-test capability on the card or provide electrical access to the input and output lines of the LSI devices. If sockets, self-test, and electrical access are all impractical, recognize that card test will be expensive.

d. All input-output signals should be TTL compatible. Interface cards which use non-TTL signal levels are normally considered "analog" cards. Testers include pull-up resistors so open collector outputs may be used. The control line for tri-state busses should be externally accessible. Input signals including clock and master clear require 20 milliamperes or less.

e. Oscillators and monostables should be implemented so that, a) they can be electrically disconnected using external logic signals, and b) paths are provided for external injection of the functions they would normally provide. Routing a signal off the board and back in via I/O pins is another way to satisfy these requirements.

f. Large feedback loops, long logic paths, and long counters (more than 8 bits) should be broken, preferably at an I/O connector or by an externally controlled logic signal. Test points can provide control inputs or signal outputs to break loops, counters, and paths. IC sockets without inserted components can provide interior test points; by the removal of jumpers they permit paths to be broken.

g. Do not use logic which depends on specific clock rates, or controlled rise and fall times, or on specific gate propagation delays.

PROMs and ROMs should have electrical access provided to their control and output lines or, if this is impossible, should be removable from sockets.

2.2 Appendices 1 and 2 contain design techniques applicable to discrete digital and digital IC circuits.(11)(13) Appendix 2 guidelines are especially applicable to compatibility with automatic digital test techniques.

2.3 Testing a nonremovable, free-running oscillator is often a problem (Figure 1).(14) One solution, if the oscillator can withstand having its output shorted to ground for several seconds, is shown in Figure 2. Table 1 describes the circuit action. Successful testing depends upon placing the 74S65 as close to the flip-flop as possible. One method is to mount the 74S65 on a small PC card soldered to a dip-clip that attaches to the flip-flop.

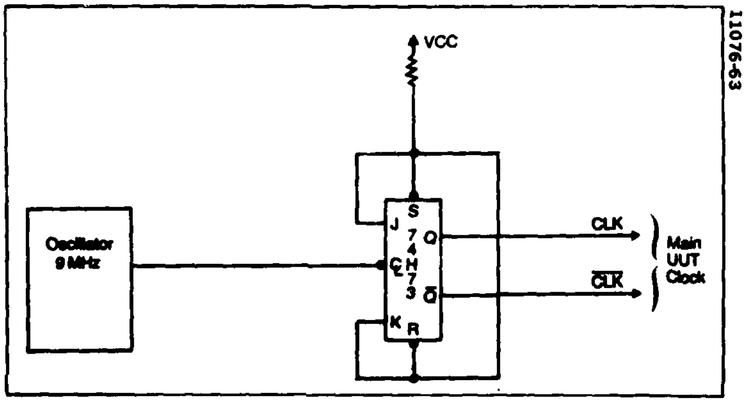


Figure 1. Problem: Nonremovable oscillator drives board clock circuitry.

11076-64

INPUTS		Previous State		'865 State	Resultant State		
Tester #2	Tester #1	Q_t	Q_t	A	Q_{t+1}	Q_{t+1}	
1	1	1	0	0	1	0	Stable -No Action
1	1	0	1	0	0	1	Stable -No Action
1	⊔	1	0	⊓	0	1	Reset Action
1	⊔	0	1	0	0	1	No Action -Stays Reset
⊔	1	1	0	0	1	0	No Action -Stays Set
⊔	1	0	1	⊓	1	0	Set-Action

Table 1. This function table describes the action of the control circuitry.

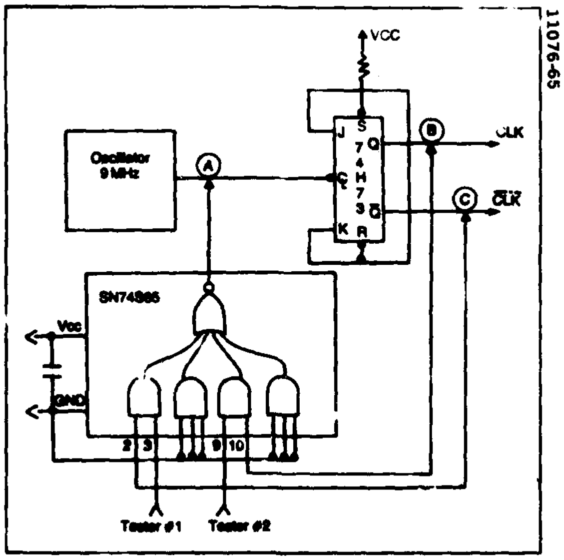


Figure 2. Solution: The 74S65 controls the oscillator using feedback from the flip-flop and control lines from the tester.

2.4 Design for "Easy" Testability(15) suggests some general rules when designing a board for testability and gives specific advice for microprocessor based designers (ROM & RAM).

2.5 "Designing MPU Boards for Testability"(16) reviews basic rules that designers should follow and discusses special problems presented by microprocessors, current strategy for testing microprocessor-based boards and self-test techniques. The board used as an example for discussion implements the IEEE STD 488-1975 digital interface and is based upon the 8080 microprocessor (contains RAM, ROM and MSI). The subjects discussed are:

a. basic design rules
b. clock-circuit control
c. access points
d. unused pins
e. testing problems presented by MPUs
f. microprocessor test strategy
g. designing a testable MPU-based board

2.6 "Self Testing VLSI Circuit Packs"(17) describes methods of implementing self-test in VLSI digital circuit packs, assesses cost factors and presents a design strategy that minimizes overhead (minimum amount of circuitry devoted solely to test), changes (minimum change in established logic design procedures) and memory (minimum storage required for testing).

2.7 Quality Criteria.

There are no firm and simple criteria for the degree of controllability to be built into a unit. Since controllability is built into the design, the contractor and government program managers should base reviews of completeness on the guidelines stated in this synopsis, applied at appropriate internal and formal design reviews.

3. Completion Criteria.

The task is considered complete upon government acceptance of the specifications prepared in Task V6.

APPENDIX V2F1 (11)

1. Simplified testing of complex sequential circuits.

In order to control a sequential circuit and test the operation of part of the circuit, additional elements can be used to force a certain state and ease testing. In the case of a latch extra inputs control the latch externally (Figure 1). Where a point is enabled by many conditions an extra input can enable this point externally (Figure 2).

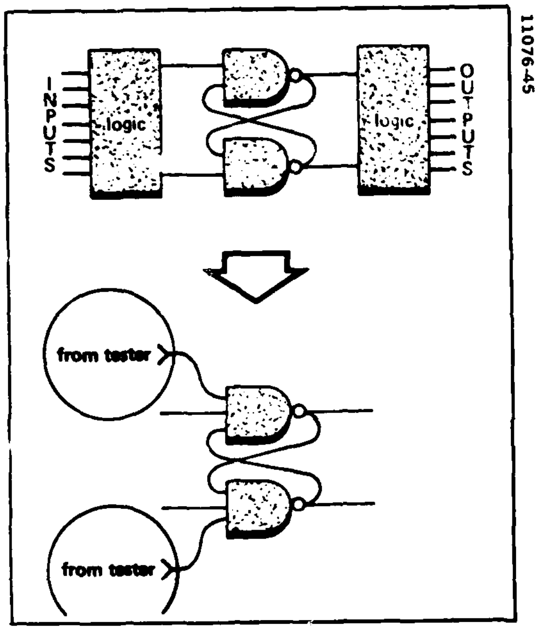


Figure 1. External Latch Control

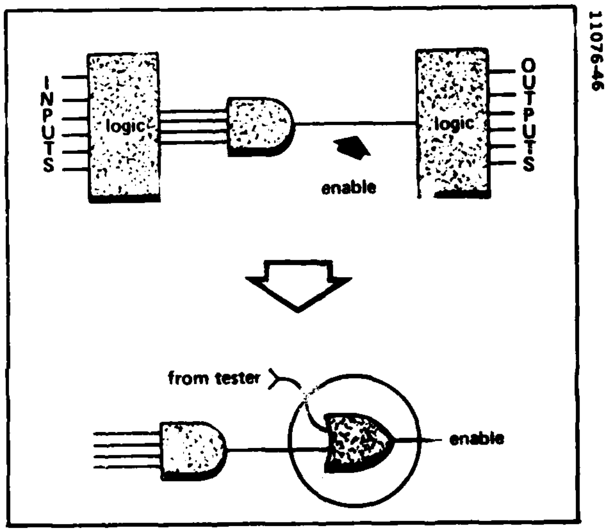


Figure 2. External Enable

2. Interrupted feedback loops.

a. Provide a means of inhibiting the clock of each memory element to break the feedback and to provide known reference points. Also add test points to each memory element in the feedback loop to improve visibility in the loop.

b. Insert an extra gate in the feedback path to interrupt feedback. The gate is controlled by a signal from the tester (Figure 3).

c. Physically break the feedback loop and bring both sides out to external pins, which are shorted by a jumper for normal operation. Removing the jumper interrupts the feedback and provides both a driving point and a sensing point (Figure 4).

d. Drive any desired test point low using the wired OR technique. When not being driven, the test point may be used for sensing.

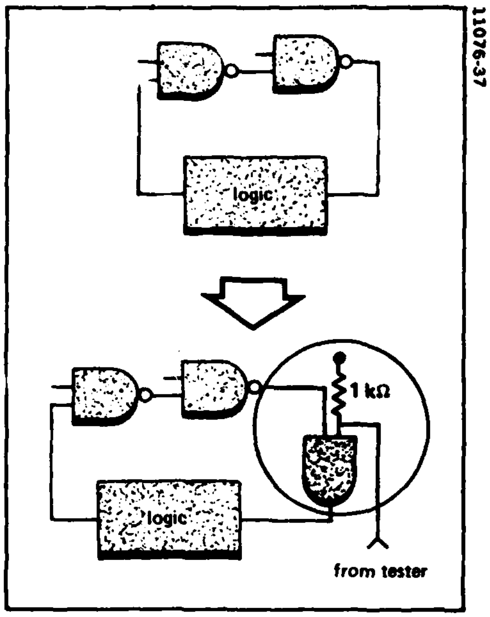


Figure 3. Gate Interrupt

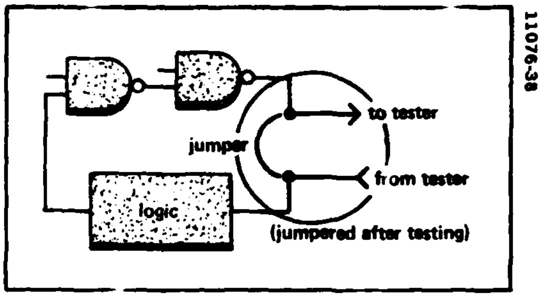


Figure 4. Jumper Interrupt

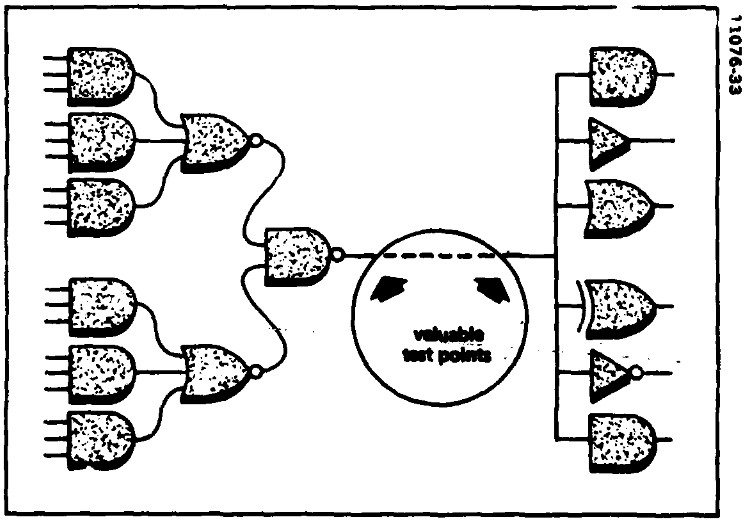


Figure 5. Test and Control Point

3. <u>Test and control point locations</u>.

a. At junctions of large fan-ins or fan-outs (Figure 5)
b. At outputs of memory elements
c. In buried logic
d. At internal branches of statically redundant logic

4. Reduced number of test patterns.

   a. Add control signals to the direct load lines of long counter chains where the counters are directly loadable.

   b. Break the cascading between counter stages to allow each stage to be clocked independently with a minimum of clock pulses. The following techniques may be used:

      - Attach a driver to pull down the cascade line, but use caution to avoid impairing the counter internally.

      - Physically interrupt the cascade lines by inserting pairs of test points which are shorted by a jumper during normal operation.

      - Add extra logic to allow counter stages to be clocked independent of the carry-out from earlier stages. For example, if the carry-out below is active in the low state, the addition of an extra gate will allow the following stage to be clocked independently (Figure 6).

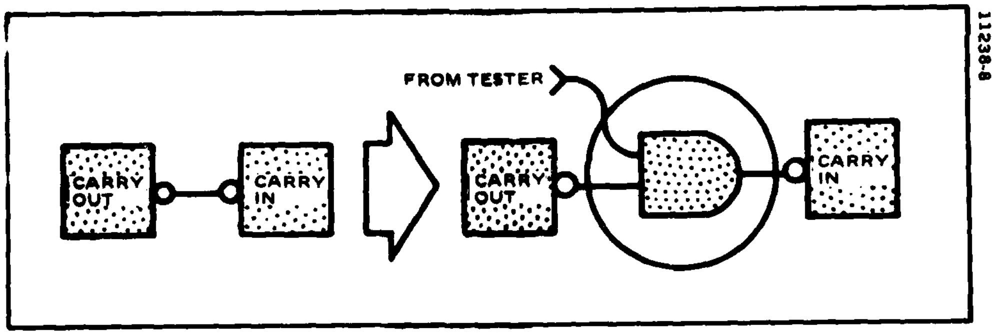


Figure 6. Counter Stage Control

APPENDIX V2F2 (13)

Appendix 2 consists of 11 self-explanatory illustrations.

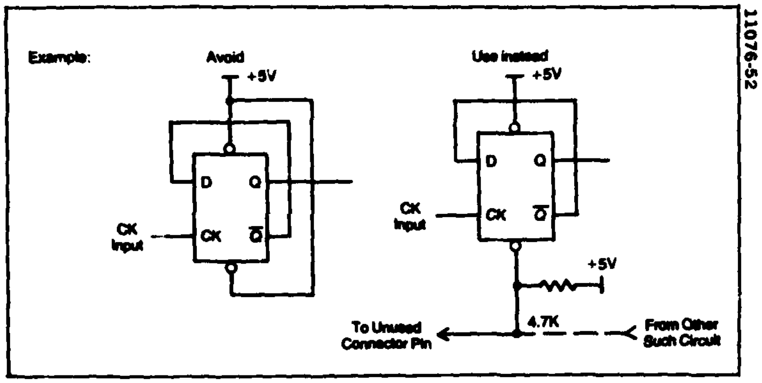


Figure 1. Insure all memory elements (flip-flops) can be preset into a known state at the beginning of the text.

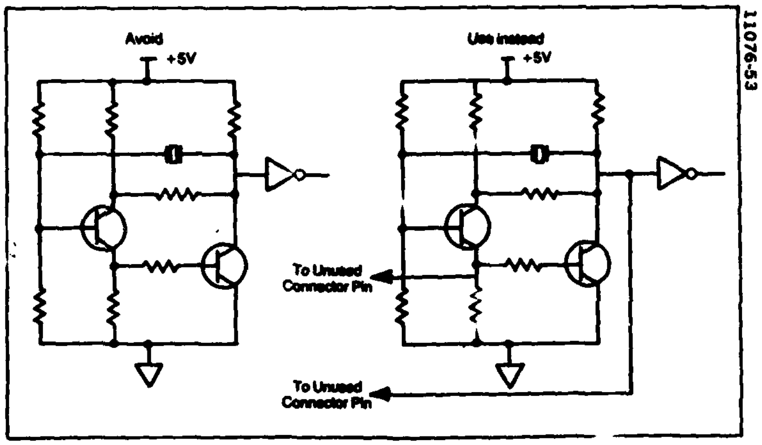


Figure 2. Allow all free running clocks to be disabled from connector and the tester clock to be inserted in place of the free running clock.

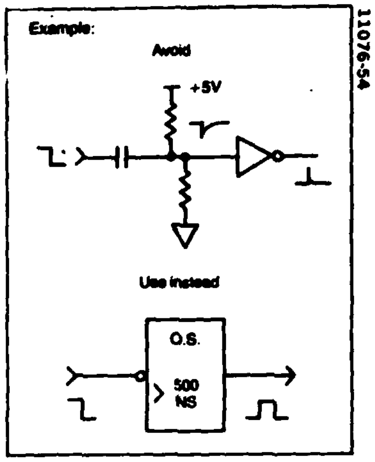


Figure 3. Avoid using circuits that produce narrow pulse widths (less than 500 ns).

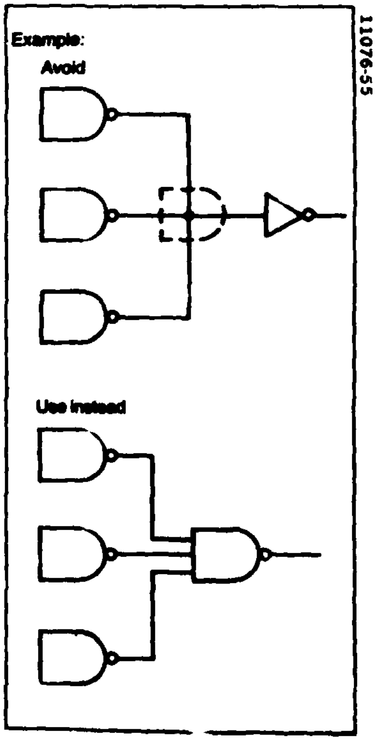


Figure 4. Avoid use of "wired AND" logic.

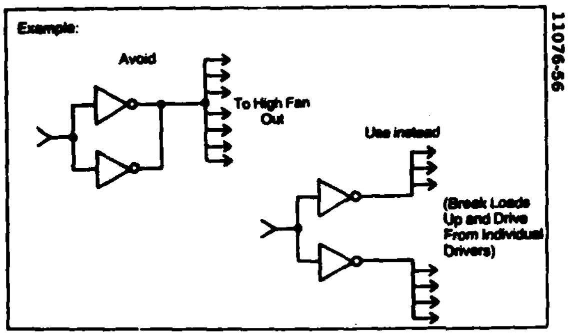


Figure 5. Avoid use of parallel gates to increase drive capability.

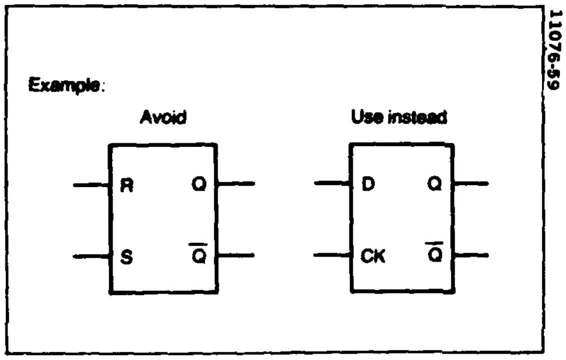


Figure 8. Avoid use of R - S flip-flops - use D type instead.

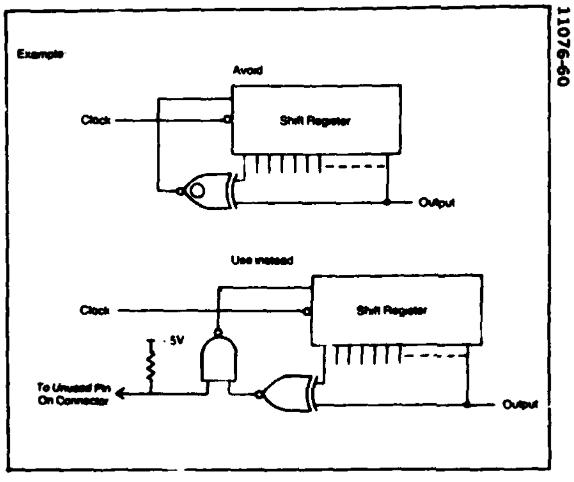


Figure 9. Provide the ability to break logical feedback loops at the connector of the board.

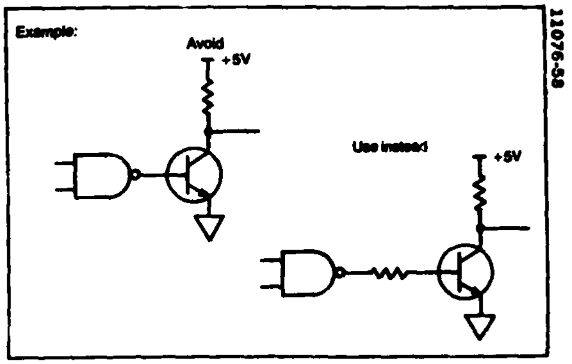


Figure 6. Avoid driving transistor bases directly from gate outputs.

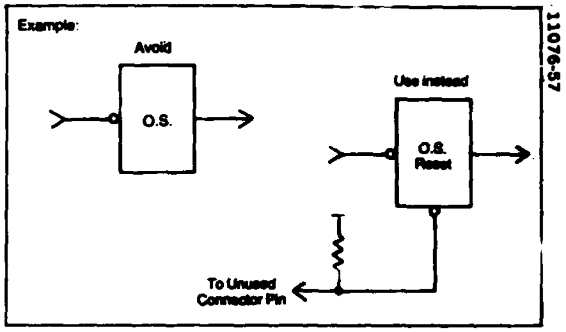


Figure 7. Utilize resetable one shots (O.S.) where possible.

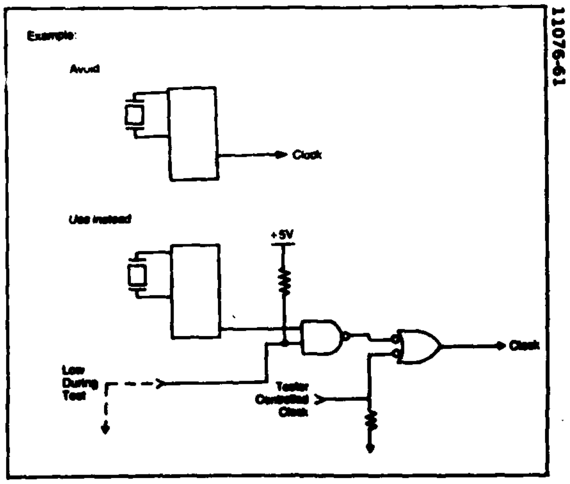


Figure 10. On CPU boards, bring the address and data bus lines to an edge connector or an internal test connector. Also, make any handshake signals available to the test programmer. Ensure the ability to operate the CPU board from an external, tester controlled clock source.

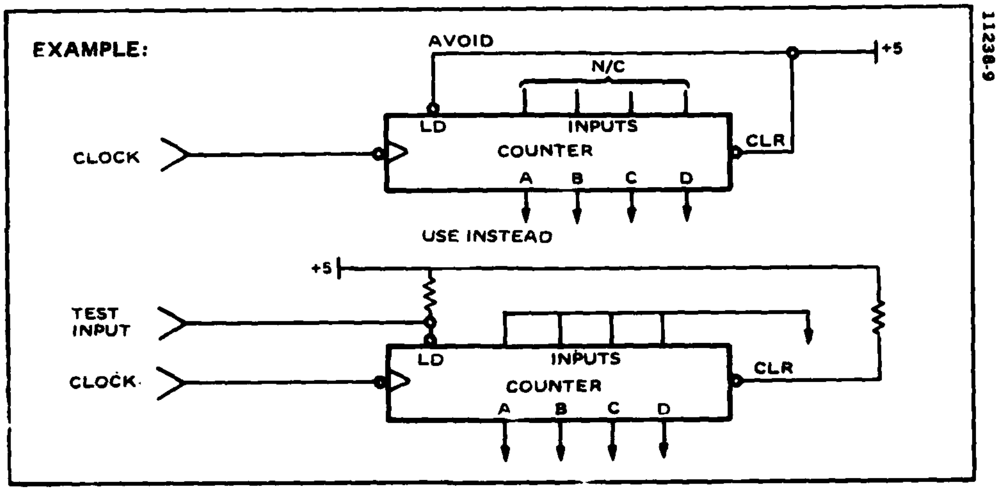


Figure 11. The Ability to Load or Preset all MSI and LSI Memory Devices Should Exist

TESTABILITY TASK COMPENDIUM
Task Reference Number V2G

PHASE: VALIDATION

FUNCTION: System Design, Testability Incorporation

TASK TITLE: Incorporate observability into design

TASK OBJECTIVE: Assure that the design provides sufficient signature data through test points, data paths, and fault isolation circuitry for the test system (BIT or ETE) to discern and isolate faults (or validly assure fault-free condition) within the module(s).

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
• Design Engineering	• Number & Placement of UUT Test Points, HW BIT Features	• Observability Requirements
• Application Software	• Software BIT Features	
• Life Cycle Cost	• Cost Guidance	• ETE Tradeoffs and Parametric Cost Data

COST TRADEOFF INTER-RELATIONSHIPS: Observability reduces the complexity of the test interface and therefore reduces the costs of TPS acquisition, and of skill levels and manhour expenditures in operations and maintenance.

TASK SYNOPSIS:

1. **Task Requirements.**(5)

1.1 The ability for external systems to perceive the function or non-function of module internal circuits is the module's observability. The observability requirement is to incorporate test points, data paths, and circuitry to provide the test system, whether BIT or ETE, sufficient signature data for failure detection and isolation within the module. The selection of physical (real) test points should be sufficient to accurately infer the value of internal nodes (virtual test points) of interest. There should be no requirement to probe internal points for organizational-level fault isolation.

2. Implementation.

2.1 The following techniques may be used to build observability into the system/module.

a. Use unused I/O pins to provide access to internal nodes otherwise unavailable.

b. Use a parity generator to achieve high observability on digital PCB's without excessive reliance on board edge connector pins as test points.(18)

c. Test Points (10)

Select test point locations for maximum access to buried nodes. Strategic placement of test points is far more important than quantity.

Critical locations for test points are:
- In feedback loops for control of important signals
- To subdivide counter chains and long sequential logic paths
- At wired AND connections and similar high ambiguity paths
- At points where high fan-out or high fan-in exists (Figure 1)
- On data bus enable signal paths
- On ROM address lines
- On chip disable bus lines

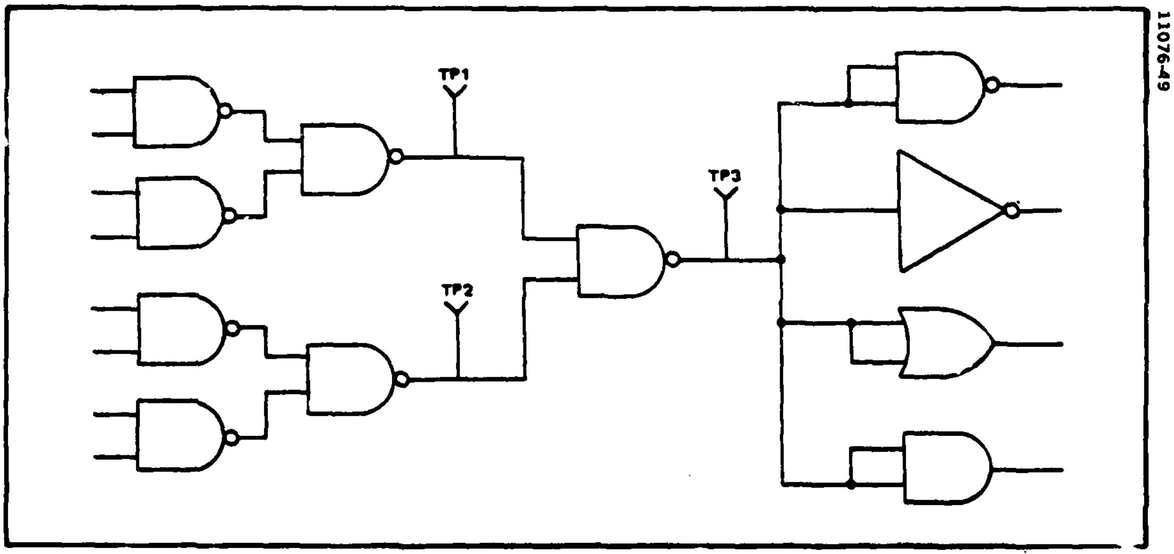


Figure 1. High Fan-Out/High Fan-In Test Point Placemen

- Between logic blocks(19)
- In circuits with hybrid (combination digital/analog) and/or redundant (fault tolerant) logic(19)

d. Use display LED's on the PCB to indicate proper operation of important circuits. Examples are: power supply voltage is OK, clock is present, or a phase-locked-loop is locked.

e. Use positive indication fault indicators and displays such that a good test always results in an ON condition. A defective indicator or display then always indicates a fault condition, either due to the input or of itself (it is faulty).

f. For a critical display, provide an alternate method of testing such as push-to-test to provide positive verification of its operation.

g. Use a multiplexer to reduce the number of edge-connector outputs for fault isolation, adjustments, and test points.(18)

h. For the situation where wired AND and Wired OR Connections are planned:(10)

Avoid the use of wired AND or OR because of the ambiguity that is created in trying to localize a fault to the specific faulty gate. Do not use wired AND(OR) with high powered TTL or Schottky TTL; they cannot be direct driven low safely for more than 1 second.

Break wired AND connections into smaller ambiguity groups. (Figure 2)

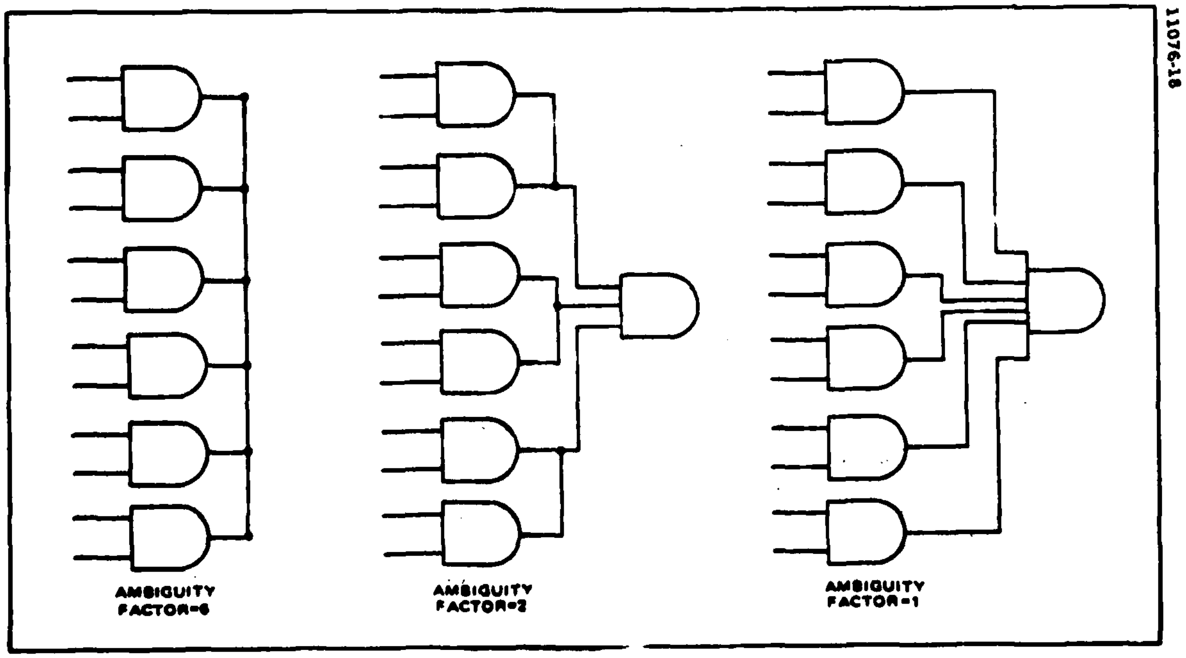


Figure 2. Redesign Gate Networks to Reduce Fault Ambiguity Factor

i. For the case where redundant circuitry is used:(10)

Avoid logically redundant circuits. A connection in a circuit is said to be redundant if no change occurs in the output of the circuit when the connection is open.

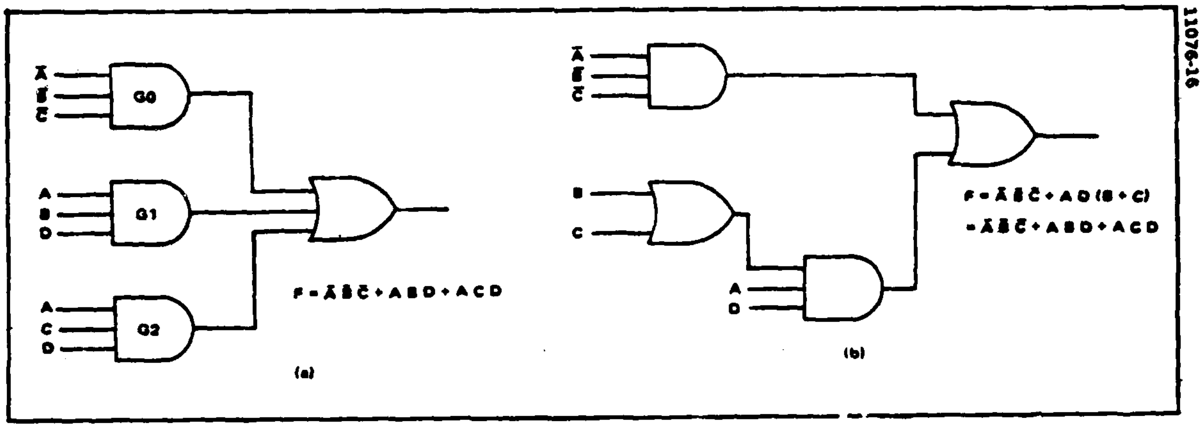


Figure 3. Provide Test Access by Redesign to Avoid Fault Redundancy

Both circuits in Figure 3 generate the same logic function. However, circuit (a) gives the same output even when the B input on gate G1 is stuck at 1, or the C input on gate G2 is stuck at 1. In either case, the output becomes F = ABC+AD, hence the fault cannot be isolated to a specific gate.

With either B or C stuck at 1, circuit (b) output also becomes F = ABC+AD, but now the fault can be isolated to the B+C gate.

2.2 One method of presenting fault isolation requirements is to relate the degree of isolation desired to component density.(20) For the most part the ratio between component density and fault isolation is relatively constant. *Figure 4 represents digital PCB's.* Isolation requirements presented in this manner do not allow design latitudes. One method of providing latitude is a non-compliance report. The key is that the responsibility for providing a testable design is placed with the designer.

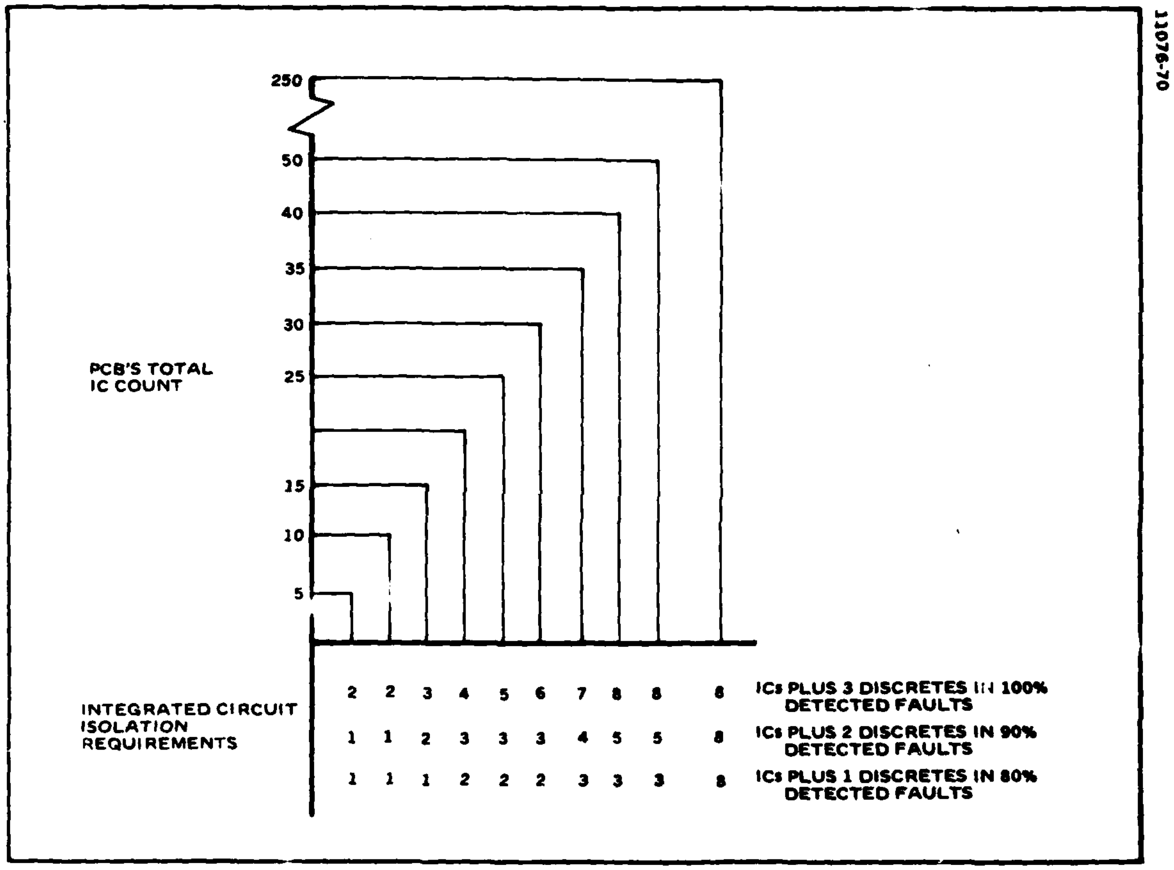


Figure 4. PCB Level Component Isolation for Digital Circuits

2.3 Quality Criteria.

a. Validation Phase Function V4, "Analyze Inherent Testability of Preliminary Design" contains synopses of the Figures of Merit used for Testability. The FOMs must be adapted and tailored for use in specific programs.

Some examples of the Figures of Merit that may be adapted to measure observability are:

$$\text{Shop Non-Ambiguity Ratio}^{(21)} \text{ (LRU Testing)*} = \frac{\text{Number of SRU's isolated directly without ambiguity}}{\text{Total number of SRU's in the LRU}}$$

$$\text{Test Thoroughness} = \frac{\text{Amount of system/equipment tested by BIT/ETE}}{\left(\text{Amount of system/equipment tested by BIT/ETE}\right) + \left(\text{Amount of system/equipment not tested by BIT/ETE}\right)}$$

$$\text{Fraction of Faults Isolated} = \frac{\text{Quantity of detected faults isolated with BIT/ETE}}{\text{Quantity of faults detected}}$$

b. Other possible measures are: (7)

- Number of components between test points
- Sensitivity of outputs with respect to internal changes
- Ratio of inputs to components
- Ratio of inputs to outputs
- Uniformity of mapping inputs to outputs, etc.

---

* LRU is a generic term that may be defined in terms of both Avionic equipment and Ground Systems equipment.
- For Avionic equipment or systems, it is Line Replaceable Unit.
- For Ground Systems equipment, it is Lowest Replaceable Unit.

LRU is defined as a unit which is designated by the plan for maintenance to be removed upon failure from a larger entity (equipment, system) in the latter's operational environment. (MIL-STD-1309-B)

A LRU is composed entirely of SRUs. A SRU is defined as follows:

SRU (Shop Replaceable Unit)- A generic term which includes all the packages within a LRU, including chassis and wiring as a unit.

c. A methodology was developed in reference (22) that evaluates the testability merits of a printed circuit board (PCB) through a "Figure of Merit" rating system that weighs the "difficult to test" and "easy to test" aspects of a circuit design. Reference (22) is condensed as an appendix to Task V4B.

3. Completion Criteria.

The task may be considered complete upon acceptance at the PDR of the observability features in the design.

TESTABILITY TASK COMPENDIUM

Task Reference Number V2H

PHASE: VALIDATION

FUNCTION: System Design, Testability Incorporation

TASK TITLE: Determine the number and placement of UUT test points

TASK OBJECTIVE: Provide design guidance for optimum placement of test points at all system/equipment indenture levels for fault detection, fault isolation and inplace calibration and alignment.

DESIGN DISCIPLINE TESTABILITY ($\bar{T}$) INTER-RELATIONSHIPS:

$\bar{T}$ INTERFACE	INPUT TO $\bar{T}$	OUTPUT FROM $\bar{T}$
● Design Engineering	● Test Point Selection	● Qualitative Testability Requirements (Initialization, Controllability, & Observability)

COST TRADEOFF INTER-RELATIONSHIPS: Optimized placement of test points reduces diagnostic software development costs and reduces test resource support costs in the deployment and operations phase.

TASK SYNOPSIS:

1. Task Requirements.

1.1 The proper placement of test points is consistent with good maintenance practices. Test points should provide for validity and simplicity in the detection and isolation of faults in addition to providing for evaluation of the functional performance status of the prime system. With proper test point placement, faults can be isolated to those assemblies or components whose alignment, repair, removal/replacement or other corrective action is feasible and consistent with the maintenance and support planning.

Controllability (Task V2F) and Observability (Task V2G) are dependent on test point arrangement and should be reviewed and considered in conducting this task of test point placement.

Test point placement needs to be treated at each testable level of the prime system hierarchical indenture, i.e., the system itself represents a Unit Under Test as does every lower level UUT.

2. Implementation.

2.1 Testability is most effectively instilled into a system when design guidelines are provided as design commences. It is inefficient for the testability (or any other -ility) engineer to take the role of a post-design critic and difficult as well as uneconomical for designers to incorporate changes to documented designs when those changes reflect design criteria which could have and should have been disclosed during or ahead of design formulation. Timeliness is therefore a principal responsibility for the testability engineer, and in all interfaces, the testability engineer should be prompt, active and assertive.

2.2 There is much literature extant relative to testing of digital circuitry, while analog testing is by inference left to textbook methods. The techniques presented here have been largely kept in the form in which extracted from the source, hence they explicitly provide mostly for digital circuitry. Analog applications may however be readily inferred.

2.3 Guidance for test point placement follows in three areas: qualitative design recommendations, samples of test point locations within wiring schemes, and figure-of-merit concepts to guide the depth of assessment of effectiveness of test point placement.

a. Qualitative design recommendations are: (5)

(1) Test points should be selected based upon failure isolation requirements and unit failure rates.

(2) Test points selected should be readily accessible for connection to ATE/ETE via operational connectors or test connectors.

(3) Test points should be chosen so that high voltage and current measurements are consistent with safety requirements of MIL-STD-454.

(4) Test point measurements should relate to a common equipment ground which can also be grounded at the ATE/ETE.

(5) Test points should be decoupled from the ATE/ETE to assure that degradation of equipment performance does not occur as a result of connections to the ATE/ETE.

(6) Test points of high voltage/current should be physically isolated from test points of low logic level signals.

(7) Test points for in-place calibration/alignment should be provided as required.

(8) Test points should be selected for maximum functional observability (see Task Reference Number V2G) and controllability (see Task V2F).

(9) Test points should be located at assembly level such that fault isolation to a module can be achieved without use of module test points or disassembly of the assembly.

b. Samples of test point locations within wiring schemes are:

(1) Use wired AND (OR) for external control connections as a means of injecting test stimuli into card under test.(10) (See Figure 1.)

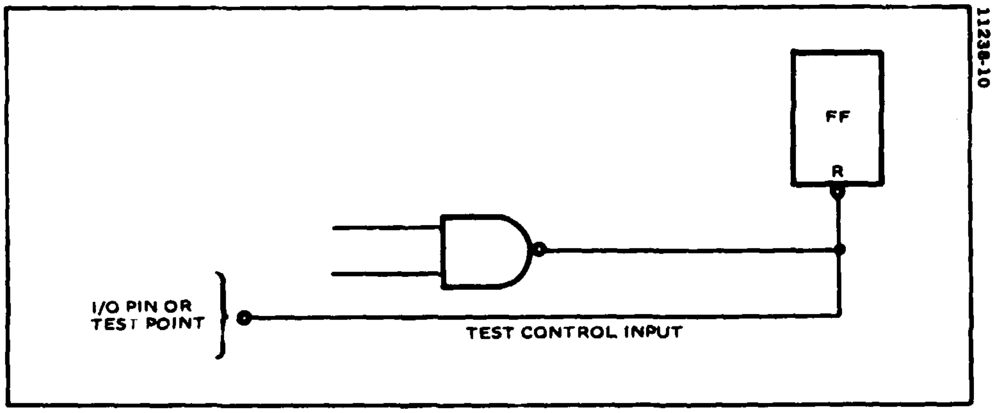


Figure 1. Wired OR for TEst Control Input

(2) Select test point locations for maximum access to buried nodes. Strategic placement of test points is far more important than quantity. (10)

(3) Place test points between logic blocks, in circuits with hybrid (combination digital/analog), in redundant (fault tolerant) circuitry, to monitor secondary d.c. power supplies and for performance monitoring of prime equipment interface signals.

(4) Important locations for test points are:(10)

- In feedback loops for control of important signals
- To subdivide counter chains and long sequential logic paths
- At wired AND connections and similar high ambiguity paths
- At any point where high fan-out or high fan-in exists (see Figure 2)
- On data bus enable signal paths
- On ROM address lines
- On chip disable bus lines

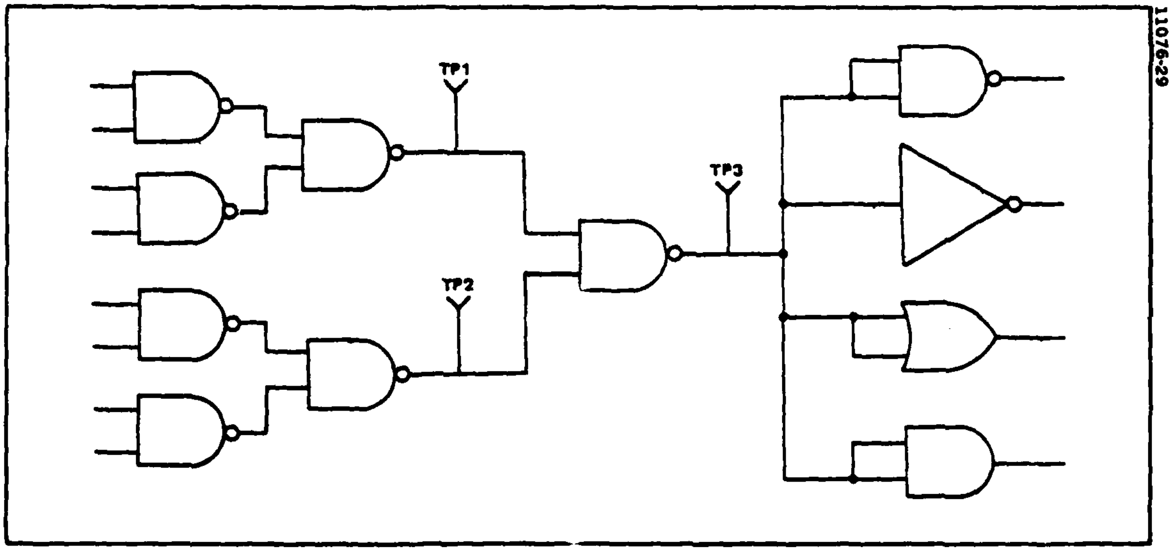


FIGURE 2. HIGH FAN-OUT/HIGH FAN-IN TEST POINT PLACEMENT

c. Figure of Merit concepts for test point placement effectiveness are:

(1) The numbers and location of the system/equipment test points should be determined to provide maximum functional controllability (task reference number V2F) and observability (task reference number V2G). The Figures of Merit for controllability and observability may be used as indicators for test point optimization.

(2) The test point number and location should be such that:

(a) the organizational level test equipment (BITE or ETE) can fault isolate the system/equipment to the replaceable assembly (subsystem/module) level without system/equipment disassembly.

(b) the intermediate level test equipment (BITE or ETE) can fault isolate the organizational level replaceable assembly to the intermediate level replaceable module (e.g., PCB/IC) without disassembly of the assembly.

(c) the depot level test equipment (ETE or ATE) can fault isolate the intermediate level module to a replaceable component (e.g., PCB/IC) without disassembly of the module.

(3) Figures-of-merit concepts that can apply to test point/ quantities are:

(a) $$\text{FOM} = \frac{\text{number of test points}}{\text{number of circuits}}$$

where both parameters are defined at the same UUT test level.

(b) $$\text{FOM} = \frac{\text{Sum of failure rates in circuits with test points}}{\text{Sum of failure rates of all circuits}}$$

(For the first FOM, the literature does not suggest a method for counting the number of circuits. A suitable quantification may be tailored to particular applications as some function of inputs and outputs (countable) for a physical entity, e.g., a subsystem, a module or circuit card or a multi-lead component.)

(4) These FOM are essentially conceptual as currently defined and no ranges of values, either possible or desirable, have been established as standards.

3. Completion Criteria.

The task of UUT test point determination is computed when the design review process actions have been completed and the government accepts the test point arrangement as designed.

TESTABILITY TASK COMPENDIUM

Task Reference Number V2I

PHASE: VALIDATION

FUNCTION: System Design, Testability Incorporation

TASK TITLE: Incorporate system level BIT

TASK OBJECTIVE: Assure that the design provides system level BIT for (1) Performance Monitoring, (2) Initial Fault Detection and Initial Fault Isolation to a major subsystem or equipment, and (3) System Verification following corrective maintenance or reconfiguration to a degraded mode.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
• System Engineering	• Boundary Between BIT/ETE	• BIT/ATE/ETE Tradeoff Analysis
• Design Engineering	• Hardware BIT Features	• HW BIT Tradeoff Analysis
• Application Software	• Test Program Architecture and Listings	• SW BIT Tradeoff Analysis

COST TRADEOFF INTER-RELATIONSHIPS: Well designed BIT improves operational availability, performance, and supportability. The costs of all test and repair resources can be substantially reduced as a result.

TASK SYNOPSIS:

1. Task Requirements.

1.1 The following requirements should be applied to the design:

a. Performance/test requirements tradeoffs.(5)

Tradeoff requirements for prime equipment performance and requirements for built-in-test and monitoring. As an example, a safety margin built into a prime equipment's output signal specification might be relaxed, with an attendant cost savings, if the accuracy and dependability of the monitoring circuits could be improved.

b. Manual/automatic test tradeoffs.

Decisions regarding the type of test equipment to be used for system monitoring and maintenance should be made based upon repair policies and overall maintenance plans. Tradeoffs should be made for test requirements at each maintenance level, considering test complexity, time to fault isolate, operational environment, logistic support requirements, development time and cost. The degree of testing automation should be consistent with the planned skill levels of the equipment operators and maintenance personnel.

c. BIT/ATE Utilization.

Establish a two-level automatic test concept which is driven by the natural break between BIT capabilities and ATE capabilities:

(1) Built-in test should be utilized to provide initial fault detection for a system or equipment and to provide initial fault isolation to a small group of modules.

(2) Off-line Automatic Test Equipment should be used to provide fault detection for a module as a Unit Under Test (UUT) and provide fault isolation to components within a module.

d. BIT/ATE Coordination.

Plan for maximum utilization of available BIT capability within a UUT in determining off-line test requirements.

e. BIT/portable tester/central shop tradeoffs.

Perform tradeoffs to determine the optimum mix of built-in test equipment, portable testers, and centralized maintenance shops to support organizational maintenance.

f. Functional partitioning for BIT.

Design the equipment such that relatively small, independent, and manageable blocks of circuitry can be defined as the basis of test derivation, test documentation and test evaluation.

g. System-level BIT.

Incorporate suitable built-in test features into the system to provide initial failure detection and to provide initial failure localization to a major subsystem or equipment.

h. System verification.

Utilize built-in test in verifying the operability of the system following a corrective maintenance action or a reconfiguration into a degraded mode.

i. Performance monitoring.

Coordinate performance monitoring requirements with BIT requirements to make optimum use of resources, as required by MIL-STD-1326.

j. Planned maintenance.

Make maximum use of available built-in test hardware and software in developing planned maintenance procedures.

k. Built-in test tradeoffs.

Incorporate a mix of built-in test, external automatic test and manual test which provides the highest level of failure resolution consistent with operational availability requirements and life cycle cost requirements. Alternate designs should be analyzed and traded off against requirements of performance, supportability, and cost to arrive at a configuration best meeting the requirements at minimum cost. General guidance in this area may be found in NAVMATINST 3960.9.

l. False Alarms.

A false alarm is defined as an indicated failure where no failure exists. The system level BIT should have a false alarm rate (frequency of occurrence of false alarms) of less than 5% of the system failure rate. Furthermore, the false alarms should occur only as a result of a failure within the system level BIT equipment.

2. Implementation.

2.1 The following guidelines may be used to incorporate system level BIT into the system design.

a. BIT includes checks made by operational software such as reasonableness checks and calibration checks. It includes continuous monitoring of integrity of data transmissions using parity and other codes, and timing of data transfers and other operations. It includes the use of end-around loop tests, microdiagnostics in control ROM, redundancy within basic control logic, and redundancy in data paths for improved fault isolation.(23)

b. BIT may be either operator or self-initiated and may display an indication visible to the operator or result in an indication detected by system confidence detection and control circuits. (23)

c. Incorporation of self-test capability into PCBs composed almost entirely of large scale ICs is a viable and powerful alternative to testing solely controlled by external test equipment. Use of self-test macro sequences initiated from and controlled by the ETE program, combines the uniqueness of a dedicated self-test with the flexibility of an externally generated test.

Among the advantages that can accrue with an internal self-test are (1) operator intervention is bypassed, (2) a fault is isolated as soon as it occurs, (3) a fault can be isolated to a replaceable component, and (4) system downtime is minimized.

Maintenance aids to incorporate into the design include built-in product-initiated self-tests and user-callable test. Self-initiated routines are automatically exercised by the product everytime some preset condition is met. A user-callable routine is exercised only when the serviceman selects it.(10)

2.2 A comparison of PCB test system characteristics using five different test scenarios is illustrated in Figure 1.(10) The figure illustrates BIT advantages.

First is built-in test, which subdivides into hardware BIT, software BIT, and the selected combination of them, called microdiagnostics. Second is drive/sense (bed-of-) nails. Third is sense only nails stimulated by edge (I/O pin) drive. Fourth is edge drive/sense. Fifth is scan/set which provides a data path out of the card via added logic and output pins separate from the functional logic; the purpose is to expose internal nodes of a chip or PCB for test purposes. The two common forms of scan are bit-serial using a shift register, and multiplex. The shift register form allows both scan of results and set(ing) of input vectors.

2.3 BIT actual failure rates have been found to exceed the rates predicted from handbooks. (24) The ratio is 1.25 actual to every predicted BIT failure.

2.4 System level BIT should meet the following quantitative criteria unless otherwise specified.

a. Fault Detection - System level BIT should detect 95% of all system failures.

b. Fault Location - System level BIT should isolate 90% of all detected failures to a single subsystem (assembly), 95% of all detected failures to no more than two subsystems, and 100% of detected failures to no more than three subsystems.

c. False Alarm - System level BIT should have a false alarm rate of less than 5% of the system failure rate.

11076-51

TEST SCENARIO					
TEST SYSTEM CHARACTERISTICS	DESIGN PCB FOR BIT USING MICRODIAGNOSTICS	DRIVE/SENSE BED-OF-NAILS	EDGE PIN STIMULUS COMBINED WITH BED-OF-NAILS SENSE ONLY	STIMULUS AND SENSE AT EDGE PINS ONLY	EDGE PIN WITH PCB DESIGNED FOR SCAN/SET
TEST TIME REQUIRED	FEW SECONDS	FEW MINUTES	FEW SECONDS	FEW SECONDS	FEW SECONDS
FAULT RESOLUTION. FAULT RESOLVED TO- COMMENTS	VERY HIGH. NORMALLY TO FAILING CHIP- ADDITIONAL CONTROL NEEDED TO RESOLVE CHIPS ON A BUS	VERY HIGH RESOLUTION OF CHIP FAILURES. FOIL NOT CHECKED WELL-PRETEST PBW AS BACKUP	HIGH RESOLUTION. TO CHIP OR BUS NODE- ALSO CHECKS FOIL.	LOW RESOLUTION. TO EXTERNAL NODE-	HIGH RESOLUTION. TO CHIP OR BUS NODE-
RELIABILITY OF TESTER	NO TESTER REQUIRED. MUST HAVE PROCESSOR ON PCB TO RUN BIT	LOW MODERATE	HIGH MODERATE	HIGH	HIGH
SOFTWARE DEVELOPMENT COST	NOMINAL	NOMINAL	HIGH	HIGH	HIGH
TEST GENERATION COST/MANPOWER AND COMPUTER TIME.	NOMINAL	NOMINAL	HIGH	VERY HIGH	HIGH
EASE OF TEST UPDATE	EXCELLENT. ONLY NEEDED TO MODIFY BUILT-IN DIAGNOSTIC PROGRAM	GOOD. INTERCONNECTION BETWEEN CHIPS IS OF MINOR IMPORTANCE. FOR NEW CHIPS ADD A NEW FUNCTIONAL TEST TO EXISTING PACKAGE	POOR. EDGE PINS USUALLY REQUIRE A WHOLE NEW TEST IF ANY PART OF THE CARD IS CHANGED.		
COST TO REPRODUCE THE TESTER	NONE. NO TESTER REQUIRED	VERY HIGH	HIGH	NOMINAL	NOMINAL
TEST DATA BASE SIZE. COST COMMENTS	SMALL. LOW, VERY INFREQUENT UPDATES. EASY TO DO USING MICRODIAGNOSTICS	SMALL TO MODERATE. NOMINAL, FREQUENT UPDATES. ONE TEST PER CHIP TYPE	LARGE. HIGH. FREQUENT UPDATES.		
DESIGN GUIDELINES AND CONSTRAINTS ON PCBUT (IMPACT ON COST AND/OR PERFORMANCE (C/P)	MUST INCLUDE BIT IN HARDWARE AND SOFTWARE. SEVERE C/P IMPACT. VERY STRONG CONSTRAINTS	BOARD LAYOUT AND PARTS LOCATION CONSTRAINTS. SLIGHT C/P IMPACT. MODERATE CONSTRAINTS	BOARD LAYOUT AND PARTS LOCATION CONSTRAINTS. SLIGHT C/P IMPACT. LOW MODERATE CONSTRAINTS	C/P IMPACT MAY BECOME PROHIBITIVE IN LARGE LSI. MILD CONSTRAINTS	SPECIFIC CONSTRAINTS TO INCORPORATE SCAN/SET INTO DESIGN. NOMINAL C/P IMPACT. STRONG CONSTRAINTS
ABILITY TO TEST AT HIGH SPEED	HIGH	HIGH	NOMINAL	NOMINAL	NOMINAL
ABILITY TO TEST CARDS CONTAINING LSI, VLSI, VVLSI	HIGH	MODERATELY HIGH	LOW	VERY LOW	MODERATE IF SCAN/SET IS ON PCB ONLY. HIGH IF SCAN/SET IS BUILT INTO LSI
CONSTRAINTS ON TEST METHODOLOGY/TEST PROCEDURE DEVELOPMENT	VERIFY THE KERNEL, THEN EXPAND ON IT	BASICALLY A SET OF CHIP TEST ROUTINES	NEED A MEANS OF AUTO TEST VECTOR GENERATION AND COMPACTION OF RESULTS-DATA FOR PCB'S WITH SIGNIFICANT QUANTITY OF LSI		

Figure 1. A Comparison of Test System Characteristics Using Five Different Test Scenarios

d. System MTBF - Any reduction in system MTBF should be offset by an increase in system maintainability (reduced MTTR).

The ratio $\frac{MTTR}{MTBF}$ for a system with BIT should be less than the same ratio for an equivalent system without system level BIT.

Standard Figures of Merit may be used as measures for system level BIT. See Validation Phase, Task Function V4.

3. <u>Completion Criteria</u>.

The task is considered complete upon government acceptance of the development specifications.

TESTABILITY TASK COMPENDIUM

Task Reference Number V3A

PHASE: VALIDATION

FUNCTION: BIT Tradeoffs

TASK TITLE: Tradeoff alternate designs of BIT, external automatic test and manual test mixes

TASK OBJECTIVE: Provide for optimized mixes of BIT/BITE/ETE at all levels of test, maintenance and repair.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering	● System Level BIT/ETE	● Qualitative $\overline{T}$ Requirements
● Design Engineering	● Performance, HW BIT	● Qualitative $\overline{T}$ Requirements
● Maintainability Engineering	● Maintenance Concept	● Merge $\overline{T}$ and $\overline{M}$ Requirements
● Support Equipment	● Selected or Available ETE vs New ETE Analysis	● Qualitative $\overline{T}$ Requirements
● Application Software	● Test Sequences, SW BIT	● Qualitative $\overline{T}$ Requirements
● Life Cycle Cost	● Cost Data Guidance	● Test Cost Data
● ILS Engineering	● Repair vs Discard Concepts and Information	● Test Equipment Options
● Program Manager		● Testability Analysis

COST TRADEOFF INTER-RELATIONSHIPS: The selection of the mix of test methods to support the prime system has major impacts on costs of acquisition and ownership. Non-recurring costs affected include prime system design, software, development test interface designs, technical data and training. Recurring acquisition costs affected include prime system, test equipment and test program sets (interface devices) together with such acquisition logistics as initial spares. Operating and support costs affected include operator and maintenance skill and labor levels and replacement training.

TASK SYNOPSIS

1. Task Requirements.

The task requires performance of iterative sets of tradeoffs.

1.1 Tradeoffs.(5)

a. Built-in-test tradeoffs. Trade off alternatives of built-in-test, external automatic test, and manual test to arrive at the optimum operational level test equipment mix.

b. BIT/Portable Tester/Central Shop Tradeoff. Trade off alternatives of BIT, portable testers, and centralized maintenance shops to arrive at the optimum organizational level test equipment mix. (In many applications, operational and organizational level testing are unified.)

c. Manual/Automatic Test Tradeoffs. Trade off alternatives of manual or automatic test equipment to arrive at the optimum test and test equipment operator requirements and personnel skill level matchups for intermediate and depot level testing. Interface with level of repair (repair vs discard) analyses.

d. Available Inventory. A search should be conducted to identify applicable test equipment within the inventory or that will be within the inventory when the system enters production stage. Adapting existing resources for use with the system is a major step in the tradeoff analysis.

2. Implementation.

2.1 Fundamental Considerations.

a. For operational level testing there are two basic test alternatives, on-line (includes BIT/BITE) and off-line. Test Equipment approaches utilizing the alternatives should be determined, examined, and a selection of the best alternative made. Considerations to be used in selection of the ETE/ATE alternative are cost, budget, schedule, technical performance, and ability to acquire suitable management capability for follow-on support. Other factors that should also be used as criteria are operational/deployment considerations, life cycle cost impacts, the system maintenance concept, development lead times to meet operational deployment, economics, reliability, maintainability, supportability, complexity of the test system including personnel skill levels, system maturity, useful life requirements, anticipated work load, UUT diversity and testability, and time constraints.(25)

b. For ground based electronics systems, offline organizational level maintenance and testing may supplement operational level testing while in most airborne electronics scenarios, there is no sharp distinction between operational and organizational level testing. Where required, the same analysis as indicated in 2.1a above may be performed to provide for selection of resources. The tradeoffs considered include portable test equipment, and iteratively treat the candidate alternatives previously considered for operational level testing.

c. Intermediate and depot level testing exist primarily to provide spares certified ready for use to the operational/organizational levels. Consequently, these levels operate under time and availability constraints differing from those imposed on operational/organizational levels, and most generally seek to optimize throughput or production rates. While therefore looking to ATE as an attractive alternative, tradeoffs at this level must also treat the candidates selected at the lower echelons, to optimally exploit such features as BIT and also to recognize tolerance requirements (depot level test tolerances must be as tight as or tighter than intermediate, which must be as tight as or tighter than organizational/operational level test tolerances).

d. The results of the tradeoff analyses conducted during the conceptual phase (Function C2) should be used in the performance of this task. Task C2A, which defined the BIT/BITE/ETE interfaces, is particularly applicable.

2.2 BIT and ETE/ATE Tradeoff Factors.

For operational/organizational level testing, some hybrid combination of BIT and ETE may often be chosen so that the test system is described as "BIT-dominant" or "ETE-dominant" rather than as a pure BIT or pure ETE. The advantages of each type are listed below: (26)

(NOTE: The reference treated BIT and ATE, adapted here to the more general concept of BIT and ETE.)

BIT DOMINANCE

- Permits instantaneous performance monitoring.
- Eases burden on operator.
- Permits fast, positive fault isolation.
- Reduces shop testing time, especially test machine occupation time.
- Reduces shop instrumentation requirements.
- Can provide specialized built-in-test capability.
- Reduces shop skill level requirements.
- Reduces shop manpower requirements.
- Reduces instruction manual and ETE software needs.
- Reduces interface cabling between UUT and ETE equipment.

- Reduces recycling time to to ETE-undetectable operational failures.
- Improves repeatability.
- Avoids circuit loading changes due to probing and prevents manually induced failures.
- Permits probing of inaccessable circuitry.
- Reduces total life cycle cost.
- Increases failure memory capability, if added.
- Permits prognostics (failure prediction).
- Reduces removal of operable units (false alarm rate).

ATE DOMINANCE

- Increases number of parameters that can be tested.
- Detects faults missed by BIT.
- Increases decision making capability (in shop).
- Permits isolation of "false alarm" causes.
- Reduces initial hardware cost (less BITE).
- Permits better "intermittent" fault isolation.

2.3 Tools and Aids for Use in the Test System Selection Process.(27)

In Table 1, the most applicable models and data banks are listed by title, function, applicable life cycle phase, reference (source or authority) and additional sources of information. The life cycle column applies to the development or acquisition phase. The basic list was compiled in 1976 and was considered most applicable to the ATE selection process at the time. Although updated to some extent for this synopsis, it is not intended as a compendium of all possible sources of aid. Additional information may be obtained from the Test Equipment Technical Support Office, Code 921, at:

Naval Ocean Systems Center
San Diego, CA 92152

Telephone: Autovon 933-6173
Commercial 714-225-6173

2.4 Recapitulation of Tradeoff Steps.

a. On-Line Test System

(1) Research literature and data banks (Table 1)
(2) Analyze approaches (selected test system/subsystem) for applicability to prime system requirements
(3) Evaluate alternatives using life cycle costing and system performance requirements and select best approach
(4) Proceed to design of system and test equipment (or modify selected test equipment).

b. Off-Line ETE/ATE

(1) Search literature and data banks (Table 1).
(2) Identify alternatives from existing inventory (including modifications of existing equipment).
(3) Evaluate alternatives using life cycle costing and system performance requirements.
(4) Select best approach and document selection.

2.5 Cost Analysis Aids.

a. BIT Life Cycle Cost Tradeoff Model. The BIT Life Cycle Cost (LCC) tradeoff model found in reference (28) provides cost delta information to assist system designers in the selection of alternate BIT concepts and design features through LCC tradeoffs. This model consists of five mathematical equations, each describing a simplified method of computing an element of LCC which is sensitive to the BIT features. The costs calculated by the model are not the total (real world) costs of the LCC elements but are relevant and useful in analyzing the incremental cost impact of alternate designs, concepts, and test subsystem features. The model is configured to use UUT level performance and design parameters. The LCC impact (delta) of the subsystem's constituent UUTs is summed by the model to provide the total subsystem incremental cost benefit or penalty.

The elements computed by the five equations are:

- RDT&E cost
- Acquisition cost
- Operation and support cost
- Availability cost
- Flight penalty cost

b. ATE Cost Drivers. Reference (29) provides analysis of economic criteria relevant to selection of ATE. The criteria were oriented to factory and service center operations and therefore have some application to depot level testing (and by inference to intermediate level). Table 2 provides condensed selection guidance factors for ATE.

TABLE 1

TOOLS & AIDS FOR USE IN THE TEST EQUIPMENT TRADEOFF AND SELECTION PROCESS

DATA BANKS

SOURCE TITLE:	GIDEP Metrology (Formerly, Secretariat, Electronic Test Equipment, Project SETE)
TYPE FUNCTION:	Data Bank-technical information, cost data, manufacturer, applications, inventory, development and production data, and evaluation data on all types of test equipment, both on-line and off-line.
APPLICABLE LIFE CYCLE PHASE:	Conceptual, Advanced Development, Full Scale Development
REFERENCES:	NAVMATINST 5200.35 MIL-STD 1556 NAVMAT Notice 5200
CONTACT FOR ADDITIONAL INFORMATION:	(Project SETE, Government-Industry Data Exchange Program Code 862) Fleet Missile Systems & Evaluation Group, Corona CA 91720; Telephone AV: 933-4351

SOURCE TITLE:	Avionic Systems Test Equipment Comparator (ASTEC)
TYPE FUNCTION:	Data Bank-Comparing and matching electronic UUTs (pin by pin) with off-line tester capabilities.
APPLICABLE LIFE CYCLE PHASE:	Advanced Development, Full Scale Development
CONTACT FOR ADDITIONAL INFORMATION:	Commanding Officer, Naval Air Engineering Center, Code SE-31, Philadelphia, PA Telephone: AV 443-4531

SOURCE TITLE:	ATE Data Bank
TYPE FUNCTION:	Data Bank-Stores off-line ATE characteristics (includes ATE building block information)
APPLICABLE LIFE CYCLE PHASE:	Advanced Development, Full Scale Development
REFERENCES:	Automatic Test Equipment Acquisition Planning Guide 1/31/74
CONTACT FOR ADDITIONAL INFORMATION:	Headquarters San Antonio Air Logistics Center Kelly Air Force Base, TX 78241 Telephone: AV 6127

TABLE 1

DATA BANKS (continued)

SOURCE TITLE: PCB Tester Data Bank (Digital and Analog)

TYPE FUNCTION: Data Bank-Compares the characteristics of PCBs to the capabilities of available military and commercial testers.

APPLICABLE LIFE CYCLE PHASE: Advanced Development, Full Scale Development

REFERENCES: RADC TR-76-106; RADC TR-79-253

CONTACT FOR ADDITIONAL INFORMATION:
Mr. James Saporito, RADC/RBET
Griffiss AFB NY 13441
Telephone: AV 587-4205; (315) 330-4205

MODELS (BIT/BITE/On-Line)

SOURCE TITLE: Built-in Test Evaluation Model (BITEM)

TYPE FUNCTION: Model-Evaluates specific BITE configurations

APPLICABLE LIFE CYCLE PHASE: Advanced Development, Full Scale Development

REFERENCES: Final Report RCA Contract N00123-73-C-1326

CONTACT FOR ADDITIONAL INFORMATION:
Naval Oceans Systems Center, Code 921
San Diego CA 92152
Telephone: AV 933-6173; (714) 225-6173

SOURCE TITLE: Determination of BIT Effectiveness Utilizing Simulation Language & General Purpose Computer

TYPE FUNCTION: Model-Predicts BIT Effectiveness for Radar & other applications.

APPLICABLE LIFE CYCLE PHASE: Advanced Development, Full Scale Development

REFERENCES: Norden Report 3676 R 1101

CONTACT FOR ADDITIONAL INFO:
Norden Division of
United Aircraft

TABLE 1

MODELS (continued)

SOURCE TITLE: An Effectiveness Study for System Formulation of Centralized Automatic Test System

TYPE FUNCTION: Model-Evaluates the effectiveness of on-line centralized ATE.

APPLICABLE LIFE CYCLE PHASE: Advanced Development

REFERENCES: DDC (AD 844872L)

CONTACT FOR ADDITIONAL INFORMATION:
Naval Ocean Systems Denter, Code 921
San Diego, CA 92152
Telephone: AV 933-6173; (714) 225-6173

SOURCE TITLE: Technique for Evaluating Avionics Built-in Test

TYPE FUNCTION: Model-Methods for evaluating the effectiveness of BIT.

APPLICABLE LIFE CYCLE PHASE: Advanced Development, Full Scale Development

REFERENCES: Contract N00019-71-C-0312

CONTACT FOR ADDITIONAL INFO:
Naval Air Systems Command
Washington DC 20361

SOURCE TITLE: A Model for Analysis for Value & Risk of a Built-in Test Equipment System

TYPE FUNCTION: Model-Measures the effectiveness for built-in test equipment.

APPLICABLE LIFE CYCLE PHASE: Advanced Development, Full Scale Development

REFERENCES: LD29196

CONTACT FOR ADDITIONAL INFO:
In-house study
Capt. Joseph Krutulis USA

TABLE 1 (continued)

SOURCE TITLE:	A Reference Library of Analytical Models and Simulations Which Can Be Used in the ATE Acquisition
TYPE FUNCTION:	ATE Model Library- Maintained to assist Navy Managers and authorized contractors in the selection and application of mathematical models relative to specific decision requirements. Provides model redommendations in order of applicability; specify input requirements, describe output interpretation, and model limitations.
APPLICABLE LIFE CYCLE PHASE:	Conceptual, Advanced Development, Full Scale Development
SOURCES OF ADDITIONAL INFORMATION:	Naval Weapons Engineering Support Activity (NAVWESA) 4204 Maylock Lane Fairfax, VA 22030

TABLE 2. ATE ECONOMY AND LONGEVITY FACTORS

HARDWARE

- Make modular for:
  - Commonality
  - Versatility
  - Provisioning
- Provide standardized:
  - Architecture
  - Bus structure
  - Packaging
  - Human interface
- Design to be:
  - Highly self-testable
  - Reliable/maintainable
  - Reconfigurable
  - Environmentally versatile
- Develop ATE hardware/software as a total system

SOFTWARE

- Use standards
- Minimize hardware/software interface problems
- Minimize use of expensive development resources
- Provide responsive/efficient development environment
- Recognize capabilities/limitations of developers
- Develop ATE software/hardware as a total system

2.6 Documentation.

The recommended selections should be documented, together with sufficient data to provide for audit to the depth required to support program decisions. Documentation should show the factors treated in analysis of the alternatives, with visibility provided especially for prime system testability, life cycle cost, test and repair times, skill levels and utilization of existing test equipment designs.

3. Completion Criteria.

3.1 This task is considered complete upon selection of the test methodology and test equipment and its acceptance at the PDR (Task V7A).

TESTABILITY TASK COMPENDIUM
Task Reference Number V4A

PHASE: VALIDATION

FUNCTION: Inherent Testability Analysis of Preliminary Design

TASK TITLE: Assess the extent of qualitative testability included in design.

TASK OBJECTIVE: Provide a formal analysis of the potential (inherent) testability of the preliminary design for feedback to and dialogue with the design engineers.

DESIGN DISCIPLINE TESTABILITY (T) INTER-RELATIONSHIPS:

$\bar{T}$ INTERFACE	INPUT TO $\bar{T}$	OUTPUT FROM $\bar{T}$
● System Engineering	● Partitioning, system BIT/ATE	● $\bar{T}$ Analysis, Continued Guidelines
● Design Engineering	● Partitioning, HW BIT, UUT Test Points, Initialization, Controllability, Parts Selected for $\bar{T}$, Observability	● $\bar{T}$ Analysis, Continued Guidelines

COST TRADEOFF INTER-RELATIONSHIPS: This task considers all the tradeoffs appropriate to achievement of a high degree of testability.

TASK SYNOPSIS:

1. **Task Requirements.**

1.1 Determine whether the following testability considerations have been included in the preliminary design.(5)

a. The equipment is physically partitioned to support the test process.

b. Each physical partition is functionally compatible with planned ETE/ATE resources.

c. Conservative timing tolerances and conservative signal tolerances are used in the design whenever possible.

d. Regular, structured designs are used whenever possible.

e. Sufficient hardware, firmware, and/or software is included to confidently drive the equipment to a known state or condition prior to running diagnostic tests.

f. The design makes maximum use of existing operational resources to support self-test in a building block fashion.

g. The equipment contains controls and displays to provide a suitable human interface for test and maintenance actions per MIL-STD-1472.

2. Implementation.

2.1 In all interfaces, the testability engineer should be prompt, active and assertive. Testability is most effectively instilled into a system if design guidelines are provided when design commences. It is inefficient for the testability (or any other -ility) engineer to take the role of a post-design critic. It is difficult as well as uneconomical for designers to incorporate changes to documented designs when those changes reflect design criteria which could have and should have been disclosed during or ahead of design formulation. Timeliness is therefore a principal responsibility for the testability engineer. The task of analyzing inherent testability will be simplified in those hardware programs where the testability engineer has been attentive to the evolving design.

2.2 Inherent testability is defined as a hardware testability design characteristic. The accumulated analyses, tradeoffs and documentation of the following tasks form the basis of the inherent testability analysis.

a.	TRN V2B	Design units to be compatible with selected or available ATE.
b.	TRN V2C	Select parts for testability; provide direct designs and test sequences.
c.	TRN V2D	Design units with testable physical partitioning.
d.	TRN V2E	Incorporate initialization into design.
e.	TRN V2F	Incorporate controllability into design.
f.	TRN V2G	Incorporate observability into design.
g.	TRN V2H	Determine number and placement of UUT test points.
h.	TRN V2I	Incorporate system level BIT.
i.	TRN V2A	General $\bar{T}$ design guidance.

2.3 Assessment of inherent testability continues through preparation of the Testability Analysis Report (Function V5) and program management/ government scrutiny in the PDR.

3. Completion Criteria.

3.1 The success criteria used for the above listed tasks also apply to this analysis, although they must be tailored and adapted. Completion of this assessment task is signified by government-approved resolution of any identified problems and by final acceptance of the testability characteristics in the course of PDR.

TESTABILITY TASK COMPENDIUM
Task Reference Number V4B

PHASE: VALIDATION

FUNCTION: Inherent Testability Analysis of Preliminary Design

TASK TITLE: Define Figure-of-Merit (FOM) for inherent testability

TASK OBJECTIVE: Establish Figures of Merit for use in evaluating and controlling the testability inherent in the prime system design.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS: None.

COST TRADEOFF INTER-RELATIONSHIPS: None.

TASK SYNOPSIS:

1. Task Requirements.

The ideal testability FOM may be defined as a numerically precise and measurable parameter that accurately evaluates the degree of testability designed into the equipment. Just as MTBF evaluates reliability and is allocable and auditable throughout the design indenture hierarchy, so too should be the Testability Figure of Merit. The FOM should be useable at any indenture level of the system which is subject to test.

1.1 There is no known and widely accepted FOM for Testability. This task then is required to screen suggested FOMs and to define the testability FOM(s) that will be used to calculate the degree of inherent testability designed into the equipment, to assess the preliminary design testability by applying the FOM(s), and to take corrective action as required before PDR.

2. Implementation.

It appears that the current state-of-the-art favors developing figures which reflect observability of faults and/or controllability of test.

2.1 The following measures of testability have been suggested in the literature without exact definitions and method of calculation. They are presented as concepts for further development by the testability engineer.

a. Observability Figures.

(1) $$\text{FOM} = \frac{\text{number of test points}}{\text{number of circuits}}$$

(2) $$\text{FOM} = \frac{\text{sum of failure rates of the circuits with test points}}{\text{sum of failure rates of all circuits}}$$

or

$$\text{FOM} = \frac{\text{sum of failure rates of all tested hardware}}{\text{total failure rate of the hardware}}$$

(3) Fault coverage = the percent of faults detected. (7)

(4) Percent of package or gates monitored. (30)

b. Combined Observability/Controllability Figures.

(1) FOM = (% of faults detected)x(% of detected faults isolated) (31)

(2) Shop Non-Ambiguity Ratio (LRU Testing) $= \dfrac{\text{Number of SRU's isolated directly without ambiguity}}{\text{Total number of SRU's in the LRU}}$ (21)

(Task Reference Number V2B contains definitions of LRU and SRU.)

(3) A FOM calculated from an inspection of logic network parameters and independent of the test generation method is: (23)

$$\text{FOM} = \sum^{n} K_j f(T_j)$$

where: K_j is a scale factor and T_j is the observable factor.

For example:

T_1 = observable test access

T_2 = measure of initialization

T_3 = measure of controllability

2.2 The following method inversely grades the degree of design response to testability requirements as a numerical value.(20) The method consists of five steps.

STEP 1. From the 44 characteristics* in Figure 1 create a table exemplified in part by Figure 2.

STEP 2. For each characteristic fill in the importance weight ("IMP WT") column of the table in Figure 2 using the data from Figure 3.

STEP 3. For each item fill in the "DESIGN RESPONSE" columns of the table in Figure 2 using the "NUMERICAL GRADE" data from Figure 4.

STEP 4. Compute the "DIFFICULTY" column of the table in Figure 2 by multiplying the "IMP WT" by the "DESIGN RESPONSE."

STEP 5. Plot the "DIFFICULTY" on Figure 1 for each item.

Figure 1, when completed, provides indications of which characteristics need most to be improved. The grading system is useful in identifying potential UUT testability problems, establishing precise task identification and defining an "inverse" FOM. (If the FOM were inverted, it would better match the conventional FOM view of the "bigger the better.")

$$\text{FOM} = \frac{\text{Sum of Difficulty Factors}}{\text{Total Number of Testable Items}}$$

2.3 A somewhat more detailed methodology which evaluates the testability merits of a PCB is developed in Appendix 1. This FOM rating system quantitatively weighs the "difficult to test" and "easy to test" aspects of a circuit design.

2.4 Computation and Action.

When developed, the FOM should be calculated, (if not calculated bottom-up) and audited by the testability and system engineering disciplines. The audit should identify design revisions that may be required. The audit results and agreed actions should be reviewed and approved or re-directed by contractor and government program management. Continuing actions should be assigned to the approximately responsible management/engineering functional people.

---

*Additional or alternative characteristics may also be used.

3. Completion Criteria.

3.1 The task is completed when there is agreement between the supplier and customer on a FOM, and the preliminary designs' inherent testability FOM has been calculated.

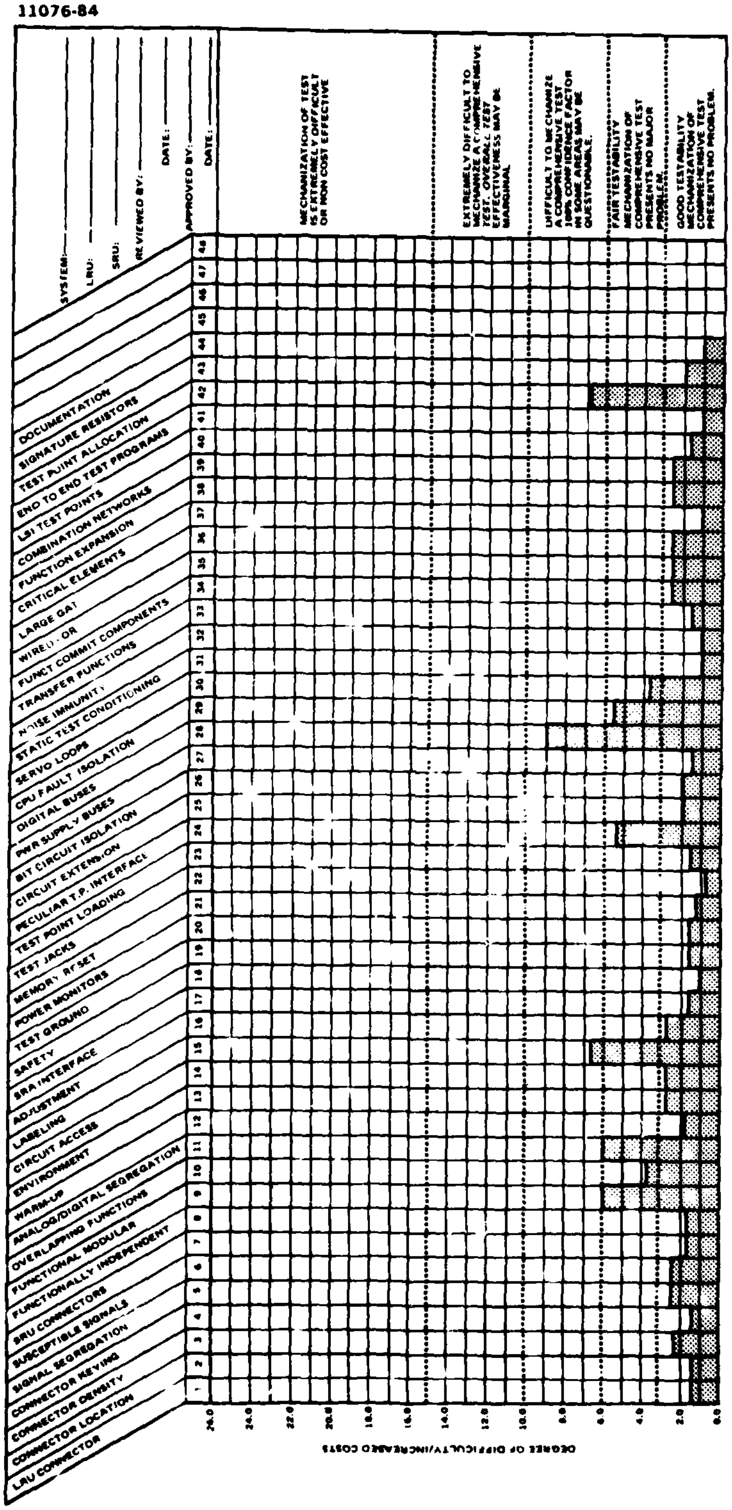


Figure 1. Testability Overview

11076-81

ITEM NUMBER	COMMON TERM	DESCRIPTION OF REQUIREMENTS	IMP WT	DESIGN RESPONSE					DIFFICULTY
				EXCEL 0	GOOD 1	FAIR 2	POOR 3	CRIT 4	
22	TEST JACKS	PREFERENCE FOR INCLUSION OF TEST JACKS FOR CONNECTION OF EXTERNAL TEST EQUIPMENT (I.E., LOGIC PROBE, LOGIC PULSER, ETC.)	0.1875			4			0.7500
24	PECULIAR TEST POINT INTERFACE	PREFERENCE FOR TEST POINT SIGNAL CONDITIONING PROVISIONS FOR UUT CRITICAL POINTS AS APPLICABLE (I.E., BUFFERS, EVENT MARKERS HIGH VOLTAGE REDUCTION, ETC.)	0.6250				9		5.6250
29	CPU FAULT ISOLATION	PREFERENCE FOR CPU BUS ACCESS, PROCESSOR ISOLATION, AND INITIALIZATION PROVISIONS	0.6250				9		5.6250
30	SERVO LOOPS	PREFERENCE FOR PROVISIONS RELATING TO POSITIVE CONTROL OF CLOSED LOOP CIRCUITS	0.9325			4			3.7300
31	STATIC TEST CONDITIONING	PREFERENCE FOR PROVISIONS TO CONTROL INTERNAL CLOCKS, OSCILLATORS, ETC.	0.9325		1				0.9325
44	DOCUMENTATION	REQUIREMENTS FOR QUALITATIVE DOCUMENTATION AVAILABILITY FOR ALL NECESSARY ELEMENTS OF TPS DEVELOPMENT	1.5625			4			6.2500

Figure 2. Testability Grading

TERM	IMPORTANCE WEIGHT	DEFINITION	EXAMPLE
CRITICAL	1.5625	A CATEGORY OF ITEMS THAT ARE REQUIRED TO PRODUCE A COST EFFECTIVE TEST	DOCUMENTATION
VERY IMPORTANT	0.9325	A CATEGORY OF ITEMS THAT ARE REQUIRED TO PRODUCE AN ACCEPTABLE LEVEL OF TEST COMPREHENSION	CONTROL AND TEST NODE SELECTION AND ACCESS
IMPORTANT	0.625	A CATEGORY OF ITEMS THAT ARE REQUIRED TO ACCOMMODATE AUTOMATIC TEST	ATE INTERFACING CONSIDERATIONS
TIME RELATED	0.375	A CATEGORY OF ITEMS THAT IMPACT TEST GENERATION TIME REQUIREMENT	UUT WARM-UP TIME
CONVENIENCE RELATED	0.1875	A CATEGORY OF ITEMS THAT RELATE TO TEST GENERATION CONVENIENCES	PROVISIONED EXTERNAL EQUIPMENT OUTLETS

11076-82

Figure 3. Importance Weighting Logic

NARRATIVE GRADE	NUMERICAL GRADE*	NARRATIVE DESCRIPTION
EXCELLENT	0	RESERVED FOR THOSE INSTANCES WHERE DESIGN RESPONSE TO TESTABILITY REQUIREMENT IS ADEQUATE IN ALL AREAS OF CONCERN
GOOD	1	RESERVED FOR THOSE INSTANCES WHERE DESIGN RESPONSE IS ADEQUATE IN AREAS OF HIGH CONCERN BUT NOT ALL AREAS OF CONCERN
FAIR	4	RESERVED FOR THOSE INSTANCES WHERE DESIGN RESPONSE IS ADEQUATE IN SOME AREAS OF HIGH CONCERN
POOR	9	RESERVED FOR THOSE INSTANCES OF LITTLE OR NO DESIGN RESPONSE TO THE TESTABILITY REQUIREMENTS IN AREAS OF HIGH CONCERN
CRITICAL	16	RESERVED FOR THOSE INSTANCES WHERE DESIGN GAVE NO CONSIDERATION TO THE TESTABILITY REQUIREMENT

*NUMERICAL GRADE = NC^2

11076-83

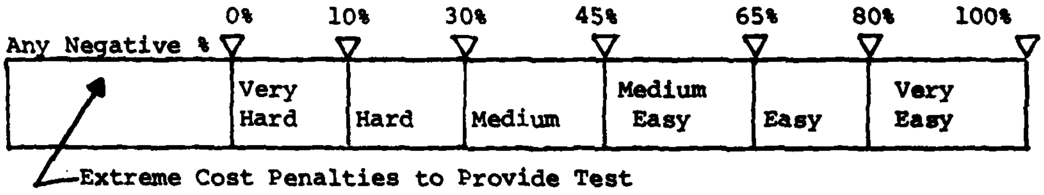
Figure 4. Design Response Grading Logic

APPENDIX V4B1[22]

PCB Testability Rating System Condensation

The PCB rating system is represented by Figure 1 (see next page). Each PCB design is rated on (i) four Basic Factors, permitting a basic weighted subscore of up to 100 percentage points and (ii) on 30 "negative" factors, each of which may subtract some finite number of percentage points from the Net Total Score. The net total score assesses PCB ease-of-testability on the following scale:

PCB RATING



To use the rating system, true representation of the actual PCB is required in the form of schematics, logic diagrams, parts lists, and specifications and internal logic of all PCB parts.

The Basic Factors

(1) B-1 rates the accessibility of circuit nodes on the PCB. To apply, count the number of input and output leads, each of which represents one accessible node. Count also the inaccessible nodes, e.g., internal part-to-part leads, counting one for each lead. The raw score for B-1 is:

$$\{(\text{number of accessible nodes}) \div [(\text{number of accessible nodes})+(\text{number of inaccessible nodes})]\} \times 100\%.$$

The count for B-1 and for subsequent factors N-3 and N-4 can be facilitated by counting the parts connected to each lead, and marking the appropriate box of the worksheet shown as figure 2. As an example, assume a board with 10 external leads each connected to 5 parts, 6 external leads each connected to 15 parts (10 marks and 6 marks respectively in the "5" and "15" boxes of the "access" half of the worksheet), 5 internal leads each connecting 10 parts and 10 internal leads each connecting 8 parts (5 marks and 10 marks in the "10" and "8" boxes in the "no access" half of the worksheet). The parts counts information will be used in N-3 and N-4.

The B-1 raw score for this example is computed from the (10+6) marks on the access half and the (5+10) marks on the no-access half of the worksheet as

$$\{(10+6)\div[(10+6)+(5+10)]\} \times 100\% = (16\div 31) \times 100\% = 52\%.$$

The raw score is weighted for "actual rating" on Figure 1 as follows on Page 7.2A-4.

FACTOR	DESCRIPTION	SCORE	POSSIBLE RATING	ACTUAL RATING	COMMENTS
B1	Percent Nodes Accessible		30%		
B2	Proper Documentation		25%		
B3	% of Sequential Ckts		25%		
B4	PCB Complexity Count		20%		
	Total Basic Score	▨	100%		
N1	Monostable Ckt		%/Inst		
N2	Counters (Pkgs & Stgs)		%/Inst		
N3	Max. No. Function Blocks/ Node (No Access)		%/Inst		
N4	Max. No. Function Blocks/ Node (Accessible)		%/Inst		
N5	Seq. Supply Voltages		10%		
N6	Non-Remov. Memories		%/Inst		
N7	Non-Remov. Buried Memory		%/Inst		
N8	Removable Complex Part		%/Inst		
N9	Non-Remov. U-Proc. VLSI		10%/Inst		
N10	Init. of Seq. Ckts		%/Inst		
N11	Ext. Loading Required		5%		
N12	Different Logic Types		%/Inst		
N13	Buried Seq. Logic		%/Inst		
N14	I/O Pins Distinguished		3%		
N15	Excess Warm-up Time		3%		
N16	Tolerance		%/Inst		
N17	High Power		%/Inst		
N18	Critical Frequency		5%		
N19	Clock Lines		20%		
N20	Ext. Test Equipment		%/Inst		
N21	Environmental		10%		
N22	Adjustments		%/Inst		
N23	Complex Signal Inputs		%/Inst		
N24	Redundant Logic		%/Inst		
N25	No. of Logic Voltages		1%/L.V.		
N26	No. of Power Supplies		1%/P.S.		
N27	Schematic Connectives		20%		
N28	I/O Pin - Schematic		5%		
N29	Dual Pin Designations		5%/Inst		
N30	Symbols on Schematic		5%		
	Total Negative Score				
	Net Total Score	▨			

FIGURE 1. PCB TESTABILITY EVALUATION SCORE SHEET

NUMBER OF CONNECTED PARTS (PKGS)

ACCESS	1	2	3	4	5	6	7	8	9	10	PCB TOTAL NODES (ACCESSIBLE) ___
	11	12	13	14	15	16	17	18	19	20	
NO ACCESS	1	2	3	4	5	6	7	8	9	10	TOTAL NODES NO ACCESS ___
	11	12	13	14	15	16	17	18	19	20	% LEADS ACCESSIBLE ___

FIGURE 2. NODE ACCESSIBILITY SCORE SHEET

SCORE %	91-100	81-90	71-80	61-70	51-60	41-50	31-40	21-30	11-20	0-10
ACTUAL RATING %	30	27	24	21	18	15	12	8	4	0

(2) B-2 rates six features of the PCB documentation with maximum weighted (actual rating) percentage points as follows:

FEATURE	% (MAX)
• Logic diagrams or schematics (of all detailed parts) provided either on overall print or as individual part specifications	4
• Detailed performance specification with signal I/O tolerances provided	8
• Truth table for each digital IC circuit type shown on schematic or on detailed part drawing provided	3
• Functional designations should be shown next to each pin number of all logic packages on the schematic	5
• Power circuits shown in a single location on the schematic and with voltages labeled	3
• Schematic shows reference to corresponding assembly print and part number of next higher assembly	2

(3) B-3 rates the sequential circuit content of the PCB. Using the schematics and the PCB parts list, add up the total number of sequential IC packages, and divide by the total of all IC packages. Count functional groups of discrete parts as equivalent to one IC. Convert the percent of sequential circuits ("Score") using scale factors below to get the actual rating. Each integrated circuit package on the schematic is counted as a single sequential or combinational circuit regardless of its individual complexity.

SEQUENTIAL CIRCUITS % SCORE	0-15	16-25	26-40	41-50	51-100
ACTUAL % RATING	25	20	10	5	0

(4) B-4 evaluates overall PCB complexity. This count is made for sequential circuits only, neglecting combinational ICs. Each circuit configuration type is scored according to the counts in the table below.

Logic Type Score Counts

Flip Flop	7
Latch	7
VLSI Chip or micro-processor	1000
4-Bit Shift Register	35
Memory Chip (n inputs)	2^n

- For complex IC circuits with sequential sections, count points for each internal combinational gate = number of input leads plus one and each inverter = 3. Then sum the counts for internal gates with the counts for the flip/flop, latch, chip and shift register logic types.

The raw score is weighted to actual rating as follows:

COMPLEXITY COUNT	Up to 300	301-500	501-800	801-1200	1201-1800	1800 +
ACTUAL RATING %	20	16	12	8	4	0

Compilation of Negative Factors

Each of 30 possible negative factors is scored in a straightforward way for entry to Figure 3.

N-1 Classify each monostable into one of three categories:

Category for Scoring	Actual Rating %
(a) Is tested by analog techniques not requiring digital ATG processing	(- 1) per instance
(b) Accessible monostable output driving sequential circuits	(- 2) per instance
(c) Inaccessible monostable output driving sequential circuits	(- 5) per instance

N-2 Multiply the number of IC packages by the number of internal sequential stages. The count of stages starts from each direct input and continues until the final stage of the counter is reached, or until a point is reached where another input can be injected.

	Scoring Factors	Actual Rating %
(a)	5 to 10 with monitor lead only	(- 2) per instance
(b)	5 to 10 not accessible	(- 3) per instance
(c)	10 or more with monitor lead only	(- 4 plus (-0.5 x (N-10)) per instance
(d)	10 or more not accessible	(-5 plus (-0.1 x (n-10)) per instance

N-3 From the bottom half of the worksheet (Figure 2) count instances where more than 3 different function blocks (circuit packages) are connected to the same (internal) wiring junction (node).

	Inaccessible Nodes	Actual Rating %
(a)	4	(-0.1) per instance
(b)	5	(-0.2) per instance
(c)	6	(-0.5) per instance
(d)	7	(-1.0) per instance
(e)	8	(-1.3) per instance
(f)	9	(-1.7) per instance
(g)	10 and higher	(-2.0) per instance

N-4 Repeat the N-3 procedure for all accessible nodes with 5 or more packages tied together.

	Accessible Nodes	Actual Rating %
(a)	5	(-0.1) per instance
(b)	6	(-0.2) per instance
(c)	7	(0.5) per instance
(d)	8	(-0.6) per instance

N-4 continued

(e)	9	(-0.8) per instance
(f)	10 and higher	(-1.0) per instance

N-5 If two or more supply voltages require a turn-on and/or turn-off sequence, assess a -10% penalty.

N-6 Any type of memory permanently wired to the PCB with all I/O leads accessible is penalized.

	Memory Size (BITS)	Actual Rating %
(a)	100K and over	(-10) per instance
(b)	32K to 99K	(-6) per instance
(c)	8K to 31K	(-4) per instance
(d)	1K to 7K	(-2) per instance

N-7 Any memory permanently wired to the PCB with one or more of its leads not connected to I/O pins is penalized:

	Memory Size (BITS)	Actual Rating %
(a)	Under 1K	(-5) per instance
(b)	$\geq$ 1K	(-10) per instance

N-8 A -1% penalty is imposed per instance of a part mounted in a socket or the equivalent and requiring extraction prior to test access.

N-9 Non-removable microprocessors, VLSI chips or other complex parts are penalized:

	Scoring Factors	Actual Rating %
(a)	All leads accessible to I/O pins	(-3) per instance
(b)	One or more leads not accessible to I/O pins	(-10) per instance

N-10 Check each sequential circuit to see if it can be initialized in two ways; using the direct set/reset inputs, and using signal input of less than 16 patterns with a clock line. Penalize each case where initialization cannot be accomplished in two ways:

	Scoring Factors	Actual Rating %
(a)	Direct set and <16 pattern reset	No penalty
(b)	Direct set but no pattern reset	(-0.05) per instance
(c)	No direct set but <16 pattern reset	(-0.1) per instance
(d)	No direct set and ≥16 pattern reset	(-2) per instance

N-11 If external loading is required by components which must be added to the Interface Device to perform test (e.g., pullup resistors), penalize as follows:

	Scoring Factors	Actual Rating %
(a)	10 resistive loads	(-2)
(b)	50 and over resistive load	(-3)
(c)	>5 Reactive Loads	(-5)

N-12 Diversity of IC type numbers is penalized:

	Scoring Factors	Actual Rating %
(a)	7 types	No penalty
(b)	10 types	(-1)
(c)	>10 types	(-1) for each additional 3 types

N-13 Start with any sequential circuit (count of 1) and count each sequential stage directly connected to one of its inputs or to one of its outputs. If an output lead from an otherwise unconnected sequential circuit is connected to the clock input of a sequential circuit in the above cluster, it should also be counted. Expand the count in all directions until all signal leads from all circuits in the cluster reach combinational circuits or a PCB input/output lead. Continue this process until all sequential circuits have been counted.

Assess penalties for each cluster of three or more sequential circuits as shown: (Do not count 2^n buried counters under this step.)

	Scoring Factors	Actual Rating %
(a)	Cluster of 3 or 4 sequential circuits	(-0.1) per instance
(b)	Cluster ≥ 5	-0.1(1+(N-5)) per instance

N-14 Input-Output pins distinguished on schematic makes tracing of signal paths easier.

	Scoring Factor	Actual Rating %
(a)	Direction arrows not different for input pins versus output pins	(-3)

N-15 Warm-up time required :o stabilize card should not exceed 3 minutes.

	Scoring Factor	Actual Rating %
(a)	Over 3 minutes	(-3)

N-16 Test equipment tolerance (if test equipment characteristics are known) is rated as follows:

	Scoring Factors	Actual Rating %
(a)	Measurement capability at least 10 times more accurate than PCB requirement	No penalty

N-16 continued

(b)	Measurement capability 3 times more accurate than PCB requirement	(-2) per instance
(c)	Measurement capability less than 3 times more accurate than PCB requirement	(-5) per instance

N-17 High power requirements are penalized:

	Scoring Factors	Actual Rating %
(a)	More than 5 amps of current required	(-5) per instance
(b)	High voltage >300Vpp	(-2) per instance
(c)	Multiple parallel pins for high current	(-1) per instance

N-18 If frequency is critical to measurement:

	Scoring Factors	Actual Rating %
(a)	Requires co-ax in Interface Device	(-5)
(b)	Over 10 MHz	(-3)
(c)	Over 4 MHz	(-2)
(d)	Over 1 MHz	(-1)

N-19 Clock lines complexity is penalized:

	Scoring Factors	Actual Rating %
(a)	One, externally controlled	(-1)
(b)	Multiphase, externally controlled	(-2)
(c)	Single clock, monitor only	(-3)
(d)	Multiple clocks, monitor only	(-5)
(e)	Inaccessible free-running clock	(-20)

N-20 Test equipment other than that contained in the automatic test equipment:

	Scoring Factors	Actual Rating %
(a)	2 power supplies or more	(-2)
(b)	Oscilloscope	(-2)
(c)	Function Generator	(-4)

N-21 Special chambers or areas required to perform test:

	Scoring Factors	Actual Rating %
(a)	Forced air, ambient or chilled	(-2)
(b)	Heat, altitude, EMI (chamber)	(-10)

N-22 Adjustments required, as for trimpots and variable capacitors, are penalized:

	Scoring Factors	Actual Rating %
(a)	per instance	(-2)
(b)	per interactive adjustment	(-4)

N-23 Interpretation by the test operator is required where complex or non-periodic waveshapes are used is penalized:

	Scoring Factors	Actual Rating %
(a)	2 coincident unusual waveforms	(-5 per instance)
(b)	1 unusual waveform	(-2 per instance)

N-24 Logic which in parallel prevents fault isolation and/or detection of individual logic failures, unless built-in-test permits fault isolation of redundant elements, is penalized:

	Scoring Factors	Actual Rating %
(a)	2 parallel logic functions - inseparable	(-2) per instance
(b)	3 and over parallel logic functions - inseparable	(-3) per instance

N-25 Excessive logic voltages are penalized:

	Scoring Factors	Actual Rating %
(a)	4	No penalty
(b)	>4	(-1 per additional logic voltage)

N-26 Excessive numbers of separate power supplies which must be supplied by the test station are penalized:

	Scoring Factors	Actual Rating %
(a)	3	No penalty
(b)	>3	(-1 each additional supply)

N-27 The aim of this factor is to guarantee that the schematic/logic diagrams do not impose hardship on the test design engineer.

	Scoring Factors	Actual Rating %
(a)	Schematic on single page	No penalty
(b)	If schematic on multiple pages with connecting leads between pages - then all interpage connectives are numbered showing other page numbers and zones	No penalty
(c)	If neither (a) nor (b) condition is met	(-20)

N-28 I/O pins located in the center of prints cause extra work for test designer.

	Scoring Factor	Actual Rating %
(a)	All I/O pins not brought to edges of schematic diagram or to a common dotted line	(-5)

N-29 Dual I/O pin designation

	Scoring Factor	Actual Rating %
(a)	If dual designation of an I/O pin is in different areas of print with no cross-reference	(-3) per instance

N-30 Only a single symbol should be used to describe a specific hardware part. Multiple symbols for identical parts make it difficult to check ATG bit propagation and to design key manual patterns to supplement tests.

	Scoring Factor	Actual Rating %
(a)	IC Logic Symbols used are not identical to detail part drawing symbols	(-5)

Totaling and Evaluation

After evaluating all factors, the Basic and Negative scores can be combined for reference to the PCB Testability Rating shown in the opening paragraph above.

TESTABILITY TASK COMPENDIUM

Task Reference Number V4C

PHASE: VALIDATION

FUNCTION: Inherent Testability Analysis of Preliminary Design

TASK TITLE: Analyze hardware/software BIT features

TASK OBJECTIVE: Document the tradeoffs made in selecting hardware BIT features using the tradeoff analysis data base for participation in a design review to determine the inclusion of suitable built-in-test features.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● Design Engineering	● HW BIT Features, Schematics, Component Listings, Chassis Diagram	● $\overline{T}$ Analysis Problem Areas and Solutions
● Application Software	● SW BIT Features, Program Lists	● $\overline{T}$ Analysis Problem Areas and Solutions
● Life Cycle Costing	● HW/SW Tradeoffs Cost Guidance	● T Analysis Costing Data

COST TRADEOFF INTER-RELATIONSHIPS: Cost relationships enter this task only indirectly, since the major cost tradeoffs occur in conjunction with task functions V2 and V3 (system design and BIT tradeoffs).

TASK SYNOPSIS:

1. Task Requirements.

1.1 The following BIT tradeoffs will be documented.(5)

a. Placement of BIT failure indicators

b. Use of standard components to implement BIT

c. Use of modular, flexible BIT designs

d. Use of active stimulus injection for BIT

e. Use of circuitry to check BIT circuitry

f. Use of circuitry to override BIT failure indications

g. Use of on-line (nondisruptive) testing and off-line (disruptive) testing.

h. Use of hardware, software and firmware BIT

1.2 Assess the inclusion of test requirements in the sizing of the memory.

a. Word allocation. Insure that sufficient words are allocated.

(1) In control memory for the storage of microdiagnostics and initialization routines.

(2) In main memory for the storage of error processing routines.

(3) In secondary memory for the storage of diagnostic routines.

b. Byte allocation. Insure that a sufficient number of bytes are assigned to each word.

(1) In control memory to achieve required controllability of hardware components.

(2) In main and secondary memory to provide for error detection/ error correction techniques, as required.

c. Protection allocation. Insure that a sufficient number of memory words are assigned to non-alterable memory resources (e.g., Read Only Memory, protected memory areas) to insure the integrity of critical test routines and data and that sufficient hardware and software redundancy exists to confidently load critical software segments.

2. Implementation.

2.1 In all interfaces, the testability engineer should be prompt, active and assertive. Testability is most effectively instilled into a system if design guidelines are provided when design commences. It is inefficient for the testability (or any other -ility) engineer to take the role of a post-design critic and difficult as well as uneconomical for designers to incorporate changes to documented designs when those changes reflect design criteria which could have and should have been disclosed during or ahead of design formulation. Timeliness is therefore a principal responsibility for the testability engineer. BIT hardware/ software design guidance should be provided by the Testability engineer for the designers' use at the outset of the design task.

2.2 The analysis involves locating testability elements on chassis drawings, schematics, functional flow diagrams, software logical flow diagrams and listings and the standard components list. The next step is to compare the baseline configuration with other possible mechanizations. For example:

a. Is the BIT result indicator located at a convenient place for sighting by the maintenance technician during maintenance?

b. Is the BIT circuitry mechanization fail safe?

c. Is memory sizing adequate for near-term increased requirements?

2.3 During the initial design stage, the proposed BIT features should be analyzed and the resulting recommendations applied to the design. This iterative process should continue until the design incorporates the required BIT features. Completion of this task occurs upon signoff of the design (at base line design).

3. Completion Criteria.

The task is considered complete upon government approval of the BIT Features Analysis in conjunction with the Preliminary Design Review.

TESTABILITY TASK COMPENDIUM

Task Reference Number V4D

PHASE: VALIDATION

FUNCTION: Inherent Testability Analysis of Preliminary Design

TASK TITLE: Conduct testability analysis of the potential UUTs in the preliminary design

TASK OBJECTIVE: Determine the extent to which testability design requirements and guidelines provided to design activities in Functions V2 and V3 have been incorporated in design, and provide guidance for subsequent detail design.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering	● Partitioning, System Level BIT/ETE, System Functional Diagram	● $\overline{T}$ Analysis Results (recommendations)
● Design Engineering	● UUT Test Points, HW BIT, Inherent $\overline{T}$ Features, Schematics, Logic Diagrams	● $\overline{T}$ Analysis Results (recommendations)
● Application Software	● SW BIT, Test Program Flow Charts and Program Lists	● $\overline{T}$ Analysis Results (recommendations)
● Reliability Engineering	● FMEA, System/Module Failure Rates	● $\overline{T}$ Analysis Results (recommendations)

COST TRADEOFF INTER-RELATIONSHIPS: Cost tradeoffs are involved only indirectly in this task, in that the task is a followup to those tasks which gave consideration to cost tradeoffs.

TASK SYNOPSIS:

1. Task Requirements.

1.1 A formal analysis and documentation of the inherent testability of the preliminary design is required as a followup to the design interfacing conducted in Function V2, Incorporation of Testability into System Design and Function V3, Performance of Test Method Tradeoffs. The characteristics to be given primary consideration are Observability, Controllability and Testability Figures-of-Merit.

The analysis may be based on a hierarchical structured format representing the prime system through a configuration listing with an associated set of qualitative and quantitative testability parameters. The format thus describes the Testability configuration of the validation phase prime system design and can be used in full scale development to improve the testability of the detail design by comparing that design with the preliminary design.

A format displaying allocated, predicted and achieved testability characteristics should be constructed for each testable configuration item (i.e., potential UUT) at each successive level of the prime system hierarchical indenture. The formats are to be used extensively by the contractor in performing design work and are convenient tools for use by the government in review. The results of the analysis should be incorporated into the Testability Analysis Report.

2. Implementation.

Functions V2 and V3 provide baseline data for use in this task. Given that the Tasks of Function V2 and V3 have been properly performed, the testability analysis may well commence on a quantitative level. If there has been no prior interface, analysis of the design for inherent T should begin on a qualitative level, then as actions are taken with system/design engineering to correct identifiable system design shortcomings, the analysis can continue to quantitative levels.

2.1 Qualitative Analysis Checklist.

If the analysis is to be initiated on a qualitative basis, the qualitative analysis should begin with a system testability checklist which would include as a minimum determining the extent of the following design features:

a. Physical partitioning
b. Functional partitioning
c. Regular structured designs
d. Initialization
e. Observability
f. Controllability
g. System compatibility with ETE resources
h. Suitable BIT features for planned maintenance concepts
i. Suitable controls and displays to provide required human interface for tests and maintenance actions.

2.2 General Notes for Detailed Quantitative Analysis.

a. The detailed analysis should include marked functional schematics which indicate the control that the test system has over the prime equipment items and the access that the test system has to the internal state of the prime items.

b. An analysis of failure modes, their effects on each item, and the ability for BIT, or ETE to fault isolate each failure should also be included.

2.3 Testability Analysis Modeling.(1)

The following excerpt is adapted from reference (1) and provides somewhat theoretical guidance for analyzing testability in a bottom-up mode which is also susceptible to computer assist. The details of the computer assist are not defined.

"Computer-assisted analysis modeling is based on the theory that the design of testable modules is straightforward when their components are themselves testable; the design of testable assemblies is straightforward when their modules are each testable; and so on up to the system level. The design process at each level is reduced to that of providing patterns to the input of each element and propagating the known failure effects through the assumed fault-free components to observable outputs.

"The test development/test measurement process starts with analysis of the several basic hardware entities for which fault simulation is performed. The stimulus/response data and testable design features required to meet specifications are ascertained through an iterative process. This lowest level of testability design and analysis may be referred to as the basic Testability Building Block (TBB). A TBB may or may not correspond to a physical or functional partition of the system.

"Higher levels in the testability design and analysis hierarchy would be defined in terms of the lower level blocks and the stimulus/response requirements for each lower level block. Any such definition may be referred to as a Testability Analysis Model (TAM). A TAM may correspond to a module, subassembly, assembly, or to any physical or functional grouping. TAMs should be chosen so as to contain approximately the same degree of complexity, independent of hierarchical level. Higher level TAMs contain more complex blocks but that complexity is transparent with respect to testability analysis. The TAM is also potentially useful as a top-down design tool for testable systems.

"Each TAM, automated or not, should permit the following analysis:

Define unambiguously the test hierarchy for this level.
Identify the response observation points (for BIT and ETE) for this level.
Evaluate the effectiveness of error monitoring circuits.
Identify any special test modes and stimulus injection points for this level.
Verify that data paths and control paths exist to stimulate lower level blocks as required, and
Verify that data paths and control paths exist to propagate the response of lower level blocks to observation points.

"The process required to support such a hierarchy of test levels and test analysis is neither a well-defined nor a well-developed process. Although the active simulation of faults is not performed beyond the lowest level (the TBB), the generation of required patterns, the calculation of responses, and the compilation of comprehensiveness data for the higher levels is not a trivial task. The process undoubtedly requires automation for analyzing systems of moderate size or larger. The process required would be closely related to the process of deductive simulation developed for digital circuits but may well be adaptable to non-digital circuits as well."

2.4 Analysis and Documentation Guidelines Notes.(1)

The following general guideline information is adapted from reference (1).

a. Figures of merit. Figures of merit for inherent testability may be developed as feasible and applied to the potential UUT represented by each Testability Analysis Model. (Refer to Validation phase Task Compendium V4B for Figures of Merit.)

b. Analysis of built-in-test features. The preliminary design should include suitable built-in-test features. The following BIT tradeoffs should be documented:

(1) Placement of BIT failure indicators
(2) Use of standard components to implement BIT
(3) Use of modular, flexible BIT designs
(4) Use of active stimulus injection for BIT
(5) Use of circuitry to check BIT circuitry
(6) Use of circuitry to override BIT failure indications
(7) Use of on-line (nondisruptive) testing and off-line (disruptive) testing
(8) Use of hardware, software and firmware BIT

c. Documentation of testability characteristics.

The qualitative description of testability features incorporated during preliminary design should be included in the preliminary Testability Analysis Report. This report serves as a single source of information on testability requirements, tradeoffs, and functional testability design for review at PDR.
(See Validation Phase Task Reference Number V5A)

3. Completion Criteria.

The task is complete upon government acceptance of the documented testability analysis results.

TESTABILITY TASK COMPENDIUM
Task Reference Number V5A

PHASE: VALIDATION

FUNCTION: Testability Analysis Report Preparation

TASK TITLE: Prepare Testability Analysis Report

TASK OBJECTIVE: Prepare a single document which represents the status and posture of the design's inherent testability as of the Preliminary Design Review.

DESIGN DISCIPLINE TESTABILITY (T) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering	● Partitioning, BIT Tradeoffs, System Level BIT/ETE	● Testability Development Specification Paragraphs
● Design Engineering	● Partitioning, Initialization, Controllability, UUT Test points, BIT for Fault Detection & Isolation, Preventive Maintenance, BIT Features and parts selection for Testability	● Testability Development Specification Paragraphs
● Maintainability Engineering	● Maintainability Program Plan and integrated logistics support plan	
● Application Software	● Direct test sequences	● CPCI Testability Paragraphs
● Reliability Engineering	● FMEA, Failure Mode Frequency	
● Life Cycle Costing	● Cost Data	
● Program Manager		● Testability Analysis Report

COST TRADEOFF INTER-RELATIONSHIPS: None directly applicable to report preparation.

TASK SYNOPSIS:

1. **Task Requirements.**

1.1 The baseline sections of the report include:(5)

a. A description of the partitioning used to enhance testability.

b. A brief functional description of each applicable hardware item.

c. An analysis of potential failure modes and effects for each item. Data on failure rates and confidence levels may be referenced from the Reliability Program, as applicable.

d. A summary of the overall maintenance concept taken from the Mainta ability Program and Integrated Logistics Support Plan and a descri tion of the overall test strategy to implement the maintenance co t including coordination between BIT and ETE.

e. A description of the test strategy to be used for each applicable item, as determined by the overall test strategy.

f. A functional description of built-in test features including hardw and software BIT.

g. A functional description of testability features, including controllability and observability considerations, based upon the preliminary Testability Analysis Model for each item.

h. A functional description of testability measurement techniques, including computer-aided analysis tools, to be used during detail design.

i. A description of the methodology used for allocating quantitative system/equipment testability requirements to the detail design of lower level items.

2. **Implementation.**

2.1 The testability analysis report serves as a single source of testability information and is directly usable by specialty engineering personnel within the originating company, support contractors and the government.

2.2 The following excerpt from the proposed Data Item Description (DID) for the Testability Analysis Report, contains the report Preparation Instructions.(1) These are additional to the baseline sections in paragraph 1.1 above.

2.2.1 The qualitative sections of the Testability Analysis Report shall include those items listed in paragraph 1.1 above.

2.2.2 The quantitative sections of the Testability Analysis Report shall include:

a. For each applicable hardware item, a description of the Testability Analysis Model including:

(1) A definition of the failure population in accordance with the specification.

(2) Identification of the test stimulus, including built-in test stimulus and external stimulus.

(3) A determination of the percentage of failures in the failure population which are detected by the test stimulus.

(4) A determination of the level of fault isolation achievable with the test stimulus.

(5) The justification for classes of failures remaining undetected or which are poorly isolated.

b. For the overall system:

(1) A description of the integration of the items and their test stimulus/response at the system level.

(2) A determination of the overall system fault coverage and level of fault isolation based upon an appropriate combination of these characteristics for each item.

(3) An estimate of developmental and recurring costs associated with design for testability, including weight, volume, and reliability penalties, and increased computer memory requirements.

(4) An estimate of developmental, production, and support savings associated with design for testability, including reduced checkout time, training, spares, and ETE costs.

2.3 The following excerpt from the proposed DID for the Testability Analysis Support Data contains the data Preparation Instructions.(1)

2.3.1 The Testability Analysis report data shall reflect each item's current configuration and include:

a. Testability Analysis Model for the item

   (1) Component characteristics

   (2) Component failure models

   (3) Component interconnection data

   (4) Test control nodes

   (5) Test observation nodes

   (6) Interrelationship of models

b. Testing Data for the item

   (1) Test stimulus

   (2) Predicted good response

   (3) Predicted fault responses

   (4) Test stimulus restrictions

   (5) Test response tolerances

   (6) Fault coverage data

   (7) Fault isolation data

3. Completion Criteria.

3.1 The Testability Analysis Report will be contractually required by CDRL/DID. The degree of acceptability can be assessed by:

a. Degree of adherence to the DID requirements.

b. Using the topic checklist in the design requirements as a gauge of completeness.

c. Having the writing services quality assurance function review the completed plan for editing principles, clarity and conciseness.

The task is completed upon government acceptance of the report.

TESTABILITY TASK COMPENDIUM
Task Reference Number V6A

PHASE: VALIDATION

FUNCTION: Development Specifications

TASK TITLE: Prepare Testability inputs to Configuration Item (CI) development specifications.

TASK OBJECTIVE: Assure that requirements for testability features are properly included in the CI specifications.

DESIGN DISCIPLINE TESTABILITY (T) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering		● System $\overline{T}$ Requirements
● Design Engineering		● $\overline{T}$ Requirements for each CI
● Support Equipment		● $\overline{T}$ Requirements for ETE Developed

COST TRADEOFF INTER-RELATIONSHIPS: There are no significant cost tradeoffs involved in the task of specification preparation. However, the contents of the specifications must be consistent with the achievement of cost effectiveness.

TASK SYNOPSIS:

1. **Task Requirements.**

It is necessary that the testability requirements for failure analysis, BIT resources, compatibility with ETE, test point access, observability, controllability, and initialization (where applicable) be included in the development specifications for each Configuration Item (CI).

1.1 **Equipment $\overline{T}$ Design Requirements.**

Refer to the tasks of functions V2, V3, and V4 for equipment design requirements and attributes. The testability requirements developed in these tasks as applicable to the CI should be included within the CI development specifications.

1.2 Tolerance Cone.

In conjunction with the specifications development, it is appropriate to establish a tolerance cone, as shown in figure 1, to preclude test inconsistencies between any of the maintenance test levels.

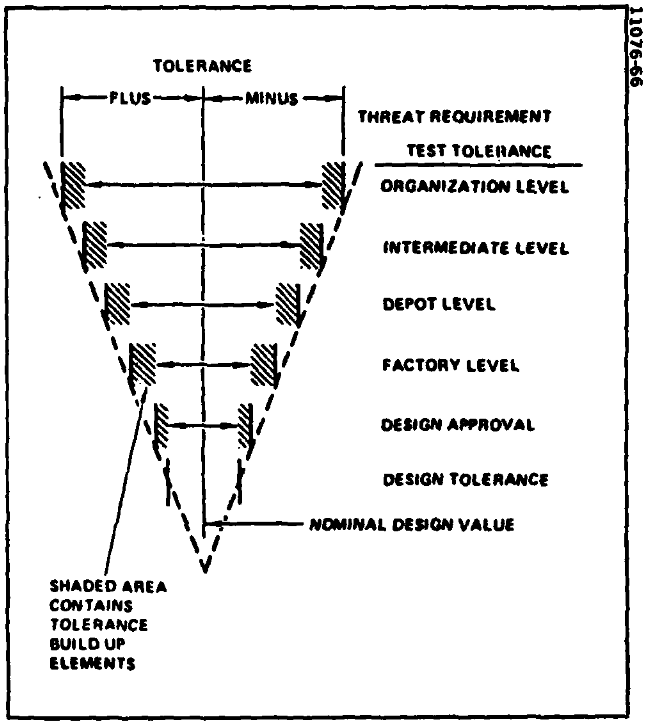


Figure 1. Tolerance Cone (32)

2. Implementation.

Given that the tasks of functions V2, V3, and V4 have been properly performed, development of the CI $\overline{T}$ requirements is relatively simplified. If there has been no prior interface, development of the CI $\overline{T}$ requirements should begin on a qualitative level. Commitments to correct identifiable shortcomings then need to be obtained from the design function. Development of the $\overline{T}$ requirements can then continue on to a quantitative level for inclusion in the CI development specifications.

2.1 Testability Characteristics for Inclusion in Specifications. (35)

The following considerations should be evaluated for applicability to, and inclusion within, the CI development specifications:

a. Definition of failure modes to be used as the basis for test design.

b. Fraction of faults detected:

  (1) percent of all faults automatically detected by BIT/ETE

  (2) percent of all faults detectable by BIT/ETE

  (3) percent of all faults detectable on-line by BIT/ETE

  (4) percent of all faults and out-of-tolerance conditions detectable by BIT/ETE

  (5) percent of all faults detectable by any means

c. Fraction of false alarms:

  (1) rate at which false indications occur (per 10^6 hours)

  (2) percent of indicated failures caused by actual failures

  (3) percent of BIT/ETE indicated failures caused by actual failures

  (4) percent of BIT/ETE fault isolations to the wrong LRU

d. Fraction of false status indications:

  (1) percent of erroneous BIT indications

e. Requirement for number of retries needed to declare a solid failure.

f. Mean fault detection time:

  (1) time to indicate a fault once it has occurred

  (2) time to detect a fault once it has occurred

g. Restrictions on built-in-test resources (allocated from system constraints).

h. Requirements for compatibility (functional, electrical, mechanical) with ETE.

i. Mean BIT/ETE running time

  (1) time to verify that a failure has occurred/or has been repaired using BIT/ETE

j. Frequency of BIT/ETE executions

  (1) time interval between BIT/ETE executions

k. Test thoroughness

  (1) percent of all equipment functions tested

l. Fault isolation resolution

(1) isolation of P_1 percent of the occurred/detected failures to X_1 UUTs, P_2 percent of the failures to X_2 UUTs and so on, with any fault isolation method.

(2) isolation of all occurred/detected faults to less than or equal to some maximum number of UUTs.

(3) isolation of P_1 percent of the occurred/detected failures to X_1 UUTs, P_2 percent of the failures to X_2 UUTs, and so on, with BIT/ETE.

(4) isolation of a specified percent of the occurred/detected failures to less than or equal to a specified quantity of UUTs at the various maintenance levels.

(5) isolation of a specified percent of the occurred/detected to less than or equal to a maximum number of plug-in modules.

(6) isolation semi-automatically to a certain percent of all occurred/detected faults down to a specified number of UUTs.

m. Fraction of faults isolated

(1) isolate a certain percent of all failures that occur and/or are detected.

(2) isolate with BIT/ETE a certain percent of all failures that occur and/or are detected.

n. Mean fault isolation time

(1) isolate a specific percent of failures that occur and/or are detected within a specified maximum time.

(2) isolate a failure down to a replaceable level, within a specified average time.

(3) isolate a failure down to a replaceable level within a specified time once the fault isolation process has been initiated.

o. Requirements for test point access.

p. Test point isolation and signal conditioning.

q. Test point density.

r. Maintenance personnel skill level

(1) all maintenance actions must be capable of being performed by a specified quantity of maintenance personnel with a specified skill level, at various maintenance levels.

(2) BIT/ETE must be designed for use by a specified minimum skill level technician.

s. BIT/ETE mean-time-to-repair

(1) mean-time-to-repair ETE

(2) mean-time-to-repair monitoring/fault isolation functions

t. BIT/ETE mean-time-between-failures

(1) mean-time-between-failures of monitoring/fault isolation functions

(2) mean-time-between-failures of ETE only

u. BIT/ETE availability

(1) monitoring/fault isolations should be operating with a specified probability of survival

v. Mean-time-to-repair

(1) system/equipment MTTR and maximum repair time

(2) system/equipment MTTR and maximum repair time at various maintenance levels

w. Availability

(1) inherent availability

(2) operational availability

x. Active memory allocated for BIT/ETE functions

(1) monitoring/fault isolation functions shall take up a specified percent of active computer memory

y. Requirements for compatibility with external status monitor.

z. Observability and controllability figures of merit.

3. Completion Criteria.

3.1 The task is completed upon incorporation of the testability goals/requirements of paragraph 1, and acceptance by the government of the CI specifications.

TESTABILITY TASK COMPENDIUM
Task Reference Number V6B

PHASE: VALIDATION

FUNCTION: Development Specifications

TASK TITLE: Prepare Computer Program Configuration Item (CPCI) development specifications

TASK OBJECTIVE: Include testability with the computer program configuration item (CPCI) requirements in the Development Specifications.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
• Design Engineering	• HW/SW Testability Features, SW Constraints Imposed by HW	• HW Testability Requirements Imposed by SW Constraints
• Application Software	• SW Testability Features	• SW Testability Requirements Imposed by HW Constraints
• System Engineering	• System SW Testability Features	• System SW Testability Requirements & HW Constraints
• Test Engineering	• Test Program Testability Features	• Test Program SW Testability Requirements & Constraints

COST TRADEOFF INTER-RELATIONSHIPS: None relevant to this task.

TASK SYNOPSIS:

1. **Task Requirements.**

1.1 The CPCI specification defines Go/No-Go, diagnostic programs, error detection (interrupt and trap capability), failure latency requirements, error processing routines (including retry, automatic error correction, diagnostic call, operator message, error logging, and immediate halt) and is consistent with the FMEA.

The CPCI development specification(s), after acceptance by the procuring activity, establish(es) the performance requirements which the CPCI must satisfy upon completion of the development phase. A CPCI development specification is required for each CPCI allocated from the system specification which established the functional baseline or from a higher level configuration item or for a non-system CI.(36) The following guidance should be applied to inputs to the CPCI specification.(4)

a. Computer Program Development Specifications. The Computer Program Configuration Item (CPCI) development specification for built-in test software (GO/NO GO and diagnostic programs) and those parts of CPCI development specifications for applications software dealing with error processing are based upon the approved preliminary design data for CI built-in test.

b. Error Detection. The application software design should include sufficient interrupt and trap capability to support the immediate processing of errors detected by concurrent built-in test hardware prior to the destruction of data bases or loss of information concerning the nature of the error. The operating system and each critical application program should contain software checks sufficient to meet failure latency requirements.

c. Error Processing. Error processing routines in the application software invoked by interrupts and traps should be designed with the full participation of hardware design engineers and test engineers. The processing to be performed (retry, automatic error correction, diagnostic call, operator message, error logging, immediate halt, and others) should be consistent with the failure modes and effects analysis. The operating system hierarchy should be designed to allow the diagnostic software sufficient control and observation of hardware components.

2. Implementation.

Implementation consists of drafting and completing the specifications.

2.1 The drafting of the CPCI specifications is consistent with MIL-STD-490 practices and relatively straightforward. Data Item Description (DID) number DI-E-30139 may be used to guide the writing of the Computer Program Development Specifications.(36) The following outline, excerpted from DID DI-E-30139, indicates the form and content:

1. General Requirements
2. Detailed Requirements
    Section 1. Scope
        1.1 Identification
        1.2 Functional Summary

Section 2 Applicable Documents
  2.1 Program Definition Documents
  2.2 Inter-Subsystem Specifications
  2.3 Military Specifications and Standards
  2.4 Miscellaneous Documents

Section 3 Requirements
  3.1 Introduction
  3.1.1 General Description
  3.1.2 Peripheral Equipment Identification
  3.1.3 Interface Identification

  3.2 Functional Description
  3.2.1 Equipment Descriptions
  3.2.2 Computer Input/Output Utilization
  3.2.3 Computer Interface Block Diagram
  3.2.4 Program Interfaces
  3.2.5 Functional Description

  3.3 Detailed Functional Requirements
  3.3.N* Introduction
  3.3.N.1 Inputs
  3.3.N.2 Processing
  3.3.N.2(a) Purpose
  3.3.N.2(b) Functional Parameters
  3.3.N.2(c) Diagrams of Geometry
  3.3.N.3 Outputs
  3.3.N.4 Special Requirements

  3.4 Adaptation

  3.5 Capacity

Section 4. Quality Assurance Provisions
  4.1 Introduction
  4.2 Test Requirements
  4.3 Acceptance Test Requirements

Section 5 Notes

Appendix A Mathematical Analysis
  A.1 Mathematical Derivations
  A.2 Alternate Method
  A.3 Summary of Equations
  A.4 Definitions of Terms
  A.5 Reference Documents

Appendix B Miscellaneous Items

3. Completion Criteria.

3.1 The CDRL, DID, and this compendium may be used to assess completeness of the CPCI specifications in their testability ramifications. The task is completed upon government acceptance of the specifications.

* N pertains to one of a number of functions, each indentured as shown.

TESTABILITY TASK COMPENDIUM

Task Reference Number V6C

PHASE: VALIDATION

FUNCTION: Development Specifications

TASK TITLE: Select ETE or prepare ETE procurement specification

TASK OBJECTIVE: Reach a timely decision for the selection of the ETE.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● Test Equipment Acquisition		● ETE Specification Testability Requirements

COST TRADEOFF INTER-RELATIONSHIPS: Not relevant to this task. Performing this task in context with its predecessor tasks in a timely manner supports cost effectiveness of the overall program.

TASK SYNOPSIS:

1. Task Requirements.

1.1 The selection or specification of ETE to support the configuration items should be accomplished prior to PDR for the prime configuration items. Units of each configuration item which are to be tested off-line (UUTs) should be identified during the validation phase. The ETE, whether selected or developed, is to be available during the full scale development phase to support Test Program Set development, testability demonstration, and maintenance of the Engineering Development Model.(5)

2. Implementation.

This task is implemented in three steps: UUT preliminary test requirements analysis, survey of existing testers and using appropriate information to prepare ETE procurement specifications. Maximum use should be made of data and information developed in prior tasks, notably those in functions V3, V4, and V5.

2.1 Test Requirements Analysis.

a. The analysis consists of documenting the UUT input-output signals listed below.

(1) Power supplies - ac, dc.
(2) Signal inputs (analog) - sinusoidal, pulse, synchro and resolver, other waveforms, time delayed.
(3) Signal inputs (digital) - serial data, parallel data.
(4) Pressure input.
(5) Signal outputs (analog) - dc voltage, ac voltage, phase angle, frequency, time period, power (average rf), other waveforms, resistance, signal distortion, synchro and resolver.
(6) Signal outputs (digital) - serial, parallel.
(7) Loads and networks - electronic elements, mechanical elements.

b. Since this task is part of the validation phase, the available UUT data may be incomplete and/or subject to change during full scale development. However, there is strong reason to develop test equipment in parallel with the prime system and for the test equipment to be available during full scale development test and evaluation.

c. A more complete checklist for requirements analysis is found in MIL-STD-1519(USAF), Preparation of Test Requirements Document. (It is not a function of this task to prepare test requirements documents.)

2.2 Tester Survey Methods.

a. The following areas should be considered in documenting commercial ATE requirements:(37)

(1) Required system-level capabilities:
  (a) Maximum
  (b) Minimum

(2) Functions to be measured on the unit under test:
  (a) Equipment Ranges
  (b) Accuracies

(3) Stimuli required:
  (a) Equipment ranges
  (b) Accuracies

(4) Reliability and maintainability required of the ATE:
  (a) Desired mean time between failure
  (b) Desired mean time to repair

(5) Power requirements:
  (a) Facility available
  (b) Tester requirements

(6) Special features such as environment, etc.

(7) Source of software development and estimated cost.

(8) Support required for such software as compilers, languages, programming manuals, etc.

(9) Maintenance concept at the requesting activity level.

(10) Utilization rate of the ATE system (per day, month, year).

(11) Systems to be tested including weapons system to be supported.

(12) Number of system units to be tested per year.

(13) Testing times for units under test by types:
  (a) Using manual test equipment
  (b) Using automatic test equipment

(14) Concurrence from applicable system manager and/or item manager relative to the use of ATE in support of their system.

(15) Impact and cost of the ATE on existing technical data.

(16) Any specific equipment to be replaced.

(17) Rationale (facts and figures) used in the cost analysis.

(18) Training required: identify by operator, maintenance and programming.

(19) Need date, with impact statement.

b. Aid in completing this task may be found in Task Reference Number 6, specifically information concerning:

- Tools and aids for use in the ATE selection process consisting of a list of data banks, models and sources of information
- ATE cost drivers
- ATE acquisition factors

2.3 Select or Specify Criteria.

a. When selecting PCB testers the usage, field or factory, should be kept in mind. Table 1 illustrates the similarities as well as the differences in importance between the two usages.(38)

b. Most PCB automatic test systems use one of two basic approaches: software simulation or hardware simulation.(39)

The advantages of software simulation include:

- The programmer has access to an exact measure of test comprehensiveness.

TABLE 1. Factory vs. Field: Importance of Test Scenario Characteristics

TEST SYSTEM CHARACTERISTICS	IMPORTANCE	
	IMPORTANCE TO FACTORY	IMPORTANCE TO FIELD MAINTENANCE
1. TESTER PERFORMANCE	HIGH	HIGH
2. FAULT RESOLUTION (TO FAILING CHIP OR FOIL)*	HIGH FOR FOIL; LOW FOR CHIPS	HIGH
3. RELIABILITY OF TESTER	NOMINAL	HIGH
4. SOFTWARE DEVELOPMENT COST	NOMINAL, EXCEPT HIGH IN SMALL INEXPENSIVE PRODUCTS OF LOW SALES VOLUME	NOMINAL, EXCEPT HIGH IN SMALL INEXPENSIVE PRODUCTS OF LOW SALES VOLUME
5. TEST GENERATION COST (MANPOWER AND COMPUTER TIME)	NOMINAL, EXCEPT HIGH IN SMALL INEXPENSIVE PRODUCTS OF LOW SALES VOLUME	NOMINAL, EXCEPT HIGH IN SMALL INEXPENSIVE PRODUCTS OF LOW SALES VOLUME
6. EASE OF TEST UPDATE	HIGH	HIGH
7. TESTER REPRODUCTION COST	NOMINAL	NOMINAL
8. TEST DATA BASE COST	LOW	HIGH
9. UNIT-UNDER-TEST LOGIC DESIGN GUIDELINES CONSTRAINTS	DESIRABLE, BUT DIFFICULT TO IMPOSE ON ENGINEERING WITHOUT HIGHER MANAGEMENT DIRECTIVES	DESIRABLE, BUT DIFFICULT TO IMPOSE ON ENGINEERING WITHOUT HIGHER MANAGEMENT DIRECTIVES
10. ABILITY TO TEST AT HIGH	HIGH	HIGH

*FOIL: THE PCB CONDUCTING PATHS FROM CHIP TO CHIP AND I/O TO CHIP.

- Except for final verification, program preparation does not require a known-good PC board.
- Automatic test generation software is available.

Disadvantages of software simulation include:
- Modeling components not in the ATE's library can prove time-consuming and costly.
- Simulation runs take a lot of time (but often can run unattended.
- PC board programming proves expensive overall, and the user forever depends on the ATE manufacturer for successive library updates.

The advantages of hardware simulation include:
- Programming does not require a description of each interconnection on the board.
- No component modeling is needed.
- Programming proves less expensive.

The disadvantages include:
- The user has no precise measure of test program comprehensiveness.
- Program preparation does require a known-good PC board.
- No automatic test generation is available.

c. The general purpose tester, such as the GR2225, is a cost effective microprocessor testing solution based on the following criteria: (40)

- The user can characterize the microprocessor to qualify and monitor vendor quality.
- High level programming permits easier programming and debugging.
- Information is available for management reports.
- Test integrity is maintained since operational uncertainties are eliminated.
- Correlation exists between field and factory testing.

d. The complexity of the new generation of ATE requires a system approach to calibration (i.e., the incorporation of calibration standards as an integral part of the system). The major advantages of this method are: (41)

- Instruments are not removed from the system.
- Since it is done with programming, human errors are reduced.
- The time required to run the calibration program is considerably less than the actual calibration time required for individual instrument.
- When the built-in standards are sent out for calibration, ATE can still be used for UUT testing.
- The system is fully calibrated.

- The system is easily adaptable to change due to IEEE 488-1975 bus.
- There is reduction in Mean Time to Repair (MTTR) due to automatic programs.

e. As the complexity and costs of manufacturing PCBs increase, development of a comprehensive board test fixture effort involves tradeoffs also.(42)

(1) Manually operated test fixtures are the least expensive to acquire, but require more time and effort to operate and generally have a limited number of probes.

(2) Vacuum fixtures can be acquired at reasonable cost, are easy to operate, and provide full access to the top of the board, but seal life expectancy and vacuum flow requirements may be adverse factors.

(3) Pneumatically operated fixtures are most costly to acquire, but overcome large probe force and test boards with components on both sides.

f. When selecting ATE, consideration should be given to the use of Graphic Display Terminals (GDT) as a fault isolation aid. Unique features and benefits of such a maintenance method are:(43)

- The GDT need not be wired to the ATE. It could be used to support many different ATE's in the same facility. Only the program tape cartridges are peculiar to the ATE.
- The GDT can also be used as a training aid to pictorially display controls, switches and functions on the ATE for completely unfamiliar personnel.
- The GDT can be used to assist system programmers, operator and even calibrators in their specific areas of responsibilities.
- All symptoms of failures and the resulting corrective actions could be written onto a tape cartridge. Recall by similar symptoms would list all previous corrective actions. Such historical data can also be consolidated and interchanged between different facilities using the same ATE.

2.4 Formulating ATE Specifications.

a. Help in formulating the specification requirements may be found in Task Reference Number V6A, CI Development Specifications. ATE specifications should be prepared per MIL-STD-490, Specification Practices.

b. The following is a checklist to prepare specifications for digital test equipment: (37)

Determine and identify:

(1) Level of assembly to be tested.
(2) Method of testing to be used.
(3) Types of digital signals to be tested.
(4) The types of digital tests to be made.
(5) The following characteristics should then be identified and defined:

(a) For Parallel Signals:

- Number of simultaneous signal input and response output lines available.
- Maximum and minimum rate that input, both repeated input and changed input, can be applied and output evaluated.
- Input and output data formatting characteristics.
- Data storage capabilities.
- Control, conditioning, and clock signals available and their characteristics.
- Stimuli characteristics and control, and response kit characteristic evaluation capabilities.
- Logic family capabilities.
- Input and output impedance characteristics.

(b) For Serial Signals:

- Stimuli characteristics including word lengths and control, bit rates and control, and word transfer characteristics.
- Data storage capabilities.
- Available control, conditioning, and clock signals and their characteristics.
- Logic family capabilities.
- Input and output impedance characteristics.
- Response evaluation characteristic of word lengths, bit characteristics and word transfer.

3. Completion Criteria.

The selection of test equipment is based upon its ability to meet UUT test requirements, its ability to be maintained and its effect upon LCC. The task is completed upon selection of the test equipment (preferably in government inventory) and/or approval of the Development Specification, Product Specification or Inventory Item Specification.

TESTABILITY TASK COMPENDIUM

Task Reference Number V7A

PHASE: VALIDATION

FUNCTION: PDR Support

TASK TITLE: Support PDR

TASK OBJECTIVE: Assure that Testability is properly represented and evaluated at the PDR.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

Coordination is appropriate with all other disciplines participating in the PDR. The inter-relationships are those of the functions and tasks which precede the PDR.

COST TRADEOFF INTER-RELATIONSHIPS: Not applicable to this task. The underlying inter-relationships are those of the functions and tasks which precede the PDR.

TASK SYNOPSIS:

1. Task Requirements.

At the Preliminary Design Review (PDR):

(1) Present the qualitative and quantitative testability predictions with support analysis (including tradeoffs, LCC and reliability data);

(2) Present status of compliance of prime system design with testability requirements;

(3) Review, revise and/or approve prime system development specifications.

1.1 The review, revision, and approval of development specifications is to be accomplished through the Preliminary Design Review (PDR) process per MIL-STD-1521. The testability design tradeoff analysis should be documented and requirements for detail design presented at the PDR. Design engineers should present qualitative and quantitative predictions of testability, with supporting analysis, and identify opportunities for further enhancement of testability through tradeoffs with performance, cost, reliability, etc. In choosing between alternate designs, preference must be given to the simplest design which meets the testability requirements.(5)

2. Implementation.

2.1 Preparation for the PDR consists of review of all generated testability materials and review of the development specifications, followed by organization of the major testability elements for concise, unambiguous presentation.

Specifications for the Preliminary Design Review process are contained in MIL-STD-1521(USAF), Technical Reviews and Audits for Systems, Equipment and Computer Programs.

The testability engineer may present summary information from the Testability Analysis Report with all necessary supporting data. This should include tradeoff results, modeling results, partitioning analysis, observability analysis, controllability analysis, initialization analysis, and BIT/ETE analysis.

The testability engineer must also participate in the review, revision, and approval of the development specifications at PDR.

2.2 The PDR may result in corrective or other actions being assigned which impact on testability. Responsibility for these actions will also be assigned. It is appropriate for the Testability discipline, while active in the developmental program, to be cognizant of actions (and their status) assigned to other disciplines or managers.

3. Completion Criteria.

Government approval of the Testability Analysis Report and the equipment/development specifications, together with approval of PDR minutes and completion of action items related to testability comprise the measure of completion.

VALIDATION PHASE BIBLIOGRAPHICAL REFERENCES

1. A Framework for Designing Testability into Electronic Systems by W.L. Keiner; TR3826, Naval Surface Weapons Center, Dahlgren Laboratory (Code K43), Dahlgren, Virginia 22448, pg. 41, 47.

2. A General Testability Design Guide by Dino J. Bonaldi; Hughes Aircraft Company internal correspondence.

3. Software Verification for Large-Scale ATE Systems by Paul P. How[illegible] Jr. and Gary R. Sadow; CH1416-7/78/0000-0076 @ IEEE 1978, pg. 76

4. MIL-STD-2076, Unit Under Test Compatability with Automatic Test Equipment, General Requirements for; 1 March 1978, pg. 6, Appendix B.

5. A Study of Testability Standardization for Electronic Systems and Equipment prepared by Testing Technology Office, NSWC and Command & Control Applications Branch, NOSC for NESC, Code 304, pg. 11, 12.

6. Design for Testability by P.L. Writer; Test Equipment Technical Support Office, Code 4050, Naval Electronics Laboratory Center, San Diego 92152; 1975 IEEE Automatic Support Systems Symposium, pg 84.

7. Testability Design Process by W.L. Keiner; Naval Surface Weapons Center, Dahlgren, VA; Industry/Joint Services Automatic Test Conference & Workshop Procedings, 3-7 Apr 78, pg 122.

8. Partitioning for Testability by Sheldon B. Akers; Journal of Design Automation & Fault Tolerant Computing, Vol I, No. 2, February 1977, pg 133.

9. AFAL-TR-79-1130, Design-for-Repair Concept Definition, Vol I: Technical Report

10. Manufacturing Methods and Technology for Digital Fault Isolation for Printed Circuit Boards prepared for U.S. Army Missile Research and Development Command, Redstone Arsenal, Alabama 35809, Contract No. DAAK 40-70-C-2090, AD-A080839, Pg. 46-67, 75, 81.

11. How to Design Logic Boards for Easier Automatic Testing and Troubleshooting prepared by General Radio Systems Division Technical Note, Pub. No. JN 794-474.

12. DPPD Digital Circuit Testability Bulletin #27, Hughes Aircraft Company, Fullerton, CA

13. Testability Guidelines by J. Mancone, Chief Production Engineer, Sierra Research Co., Buffalo NY; Electronics Test, March 1979, pg. 14.

14. Defeating an 'Undefeatable' Oscillator by James E. Scharf, Henckles, Haas & Brown, Upper Saddle River, NJ; Electronics Test, March 1979, pg. 21.

15. Design for "easy" Testability; Automated Testing for Electronics Manufacturing; Electronics Test, March 1979, pg. 25.

16. Desiging MPU Boards for Testability by Gene Foley, Teradyne, Boston, MA; Electronics Test, January 1979, pg. 48.

17. Self-Testing VLSI Circuit Packs by R.P. Davidson, Bell Laboratories, Murray Hill, NJ; Electronics Test, July 1980, pg. 34.

18. Improving PCB Testability by Norman D. Megill, Vice President, Production Services Corp., Waltham, MA; Electronics Test, January 1980, pg. 18

19. Visibility - A Key to Good Testability; Electronics Test, May 1980, pg. 128.

20. The Quantitative Measure of Testability by Conrad T. Wood, Aircraft Division, Northrop Corporation; CH1488-6/79/0000-286 @ 1979 IEEE, pg. 286.

21. Availability/Operational Readiness Test Subsystem Cost Tradeoffs, RADC-TR-80-182, pg. 32.

22. An Objective Printed Circuit Board Testability Design Guide and Rating System, RADC-TR-79-327, pg. 60.

23. Testability Measures by William Keiner & Richard West, AUTOTESTCON '77, November 2, 3, 4, 1977 @ 1977 IEEE, pg. 49.

24. Built-In-Test and External Tester Reliability Characteristics, RADC-TR-80-32, pg. 71.

25. ATE System Acquisition for E-3A Sentry (AWACS) by R.D.P. Duncan, J.H. Wilson, and R.R. Shellenbach; CH1488-6/79/000-0365 @ 1979 IEEE pg. 365.

26. Built-In-Test (BIT) Design Guide, NAVMATINST 3960.9A dtd 19 September 1979, pg. 2-14.

27. Procedure for the Selection of Automatic Test Equipment (ATE); August 1974, ATEMAT (MAT03T) Headquarters, Naval Material Command pg. 26.

28. Design Guidelines and Optimization Procedures for Test Subsystem Design, RADC-TR-80-111, pg. 180.

29. A Cost Effective Approach to ATE by G.A Drown, Rockwell International Corporation, Collins Avionics & Missiles Group; CH1488-6/79/0000-0229 @ 1979 IEEE, pg. 229.

30. A Study of a Standard Built-In-Test Circuit by J.B. Clary, Research Triangle Institute, Research Triangle Park, North Carolina 27709; Industry/Joint Services Automatic Test Conference & Workshop Proceedings, San Diego, April 1978, pg. 131.

31. 1-B Automatic Test Program Generation Recommendations by W.A. Plice; Industry/Joint Services Automatic Test Conference & Workshop Proceedings, San Diego, 3-7 Apr 78, Pg. 532.

32. Paper No. 5 Imperial Rules for Testability Based on Observation of Network Parameters by Richard Hartman; Industry/Joint Services Automatic Test Conference & Workshop Proceedings, San Diego, Apr 78, pg. 69.

33. Avionics Design for Testability by J.M. Colebank, Lockheed-California; NAECON '76 RECORD @ 1976 IEEE, pg. 223.

34. Availability/Operational Readiness Test Subsystem Cost Tradeoffs RADC-TR-80-182, pg. 16.

35. BIT/External Test Figures of Merit and Demonstration Techniques, RADC-TR-79-309, pg. 8.

36. Computer Program Product Specification, DI-E-30139.

37. Automatic Test Equipment Acquisition Planning Guide prepared and maintained by SA-ALC/MMI, Kelly AFB, TX 78241, pg. 3-B-1.

38. Future Testing of Large LSI Circuit Cards by J.H. Stewart, Sperry Univac, Roseville, Minnesota; source unknown, pg. 6.

39. Guidelines for ATE Selection, Software vs Hardware Simulation by J.G. Mancone, Chief Manufacturing Engineer, Sierra Research Corp., Buffalo, NY; Electronics Test, September 1979, pg. 36.

40. Economic Realities of Testing Microprocessors by J.I. Healy, Fairchild Instrumentation and Systems Group, Wiesbaden, West Germany; Semiconductor Test Symposium @ 1977 IEEE, pg. 47.

41. Systems Approach to Calibration of Automatic Test Equipment (ATE) by R. Shak and E. Land; Industry/Joint Services Automatic Test Conference & Workshop Proceedings, San Diego, April 78, pg. 326.

42. Test Fixture Selection Criteria by V. Kubert and P. Seuffert, Ostby and Barton, Warwick, RI; Electronics Test Introductory Issue, pg. 4.

43. Calibration, Repair, Software Support; Responsibilities by R.B. England, Senior Quality Assurance Specialist, General Dynamics Pomona Division, Pomona, CA; Industry/Joint Services Automatic Test Conference & Workshop Proceedings, San Diego, CA, April 78, pg. 324.

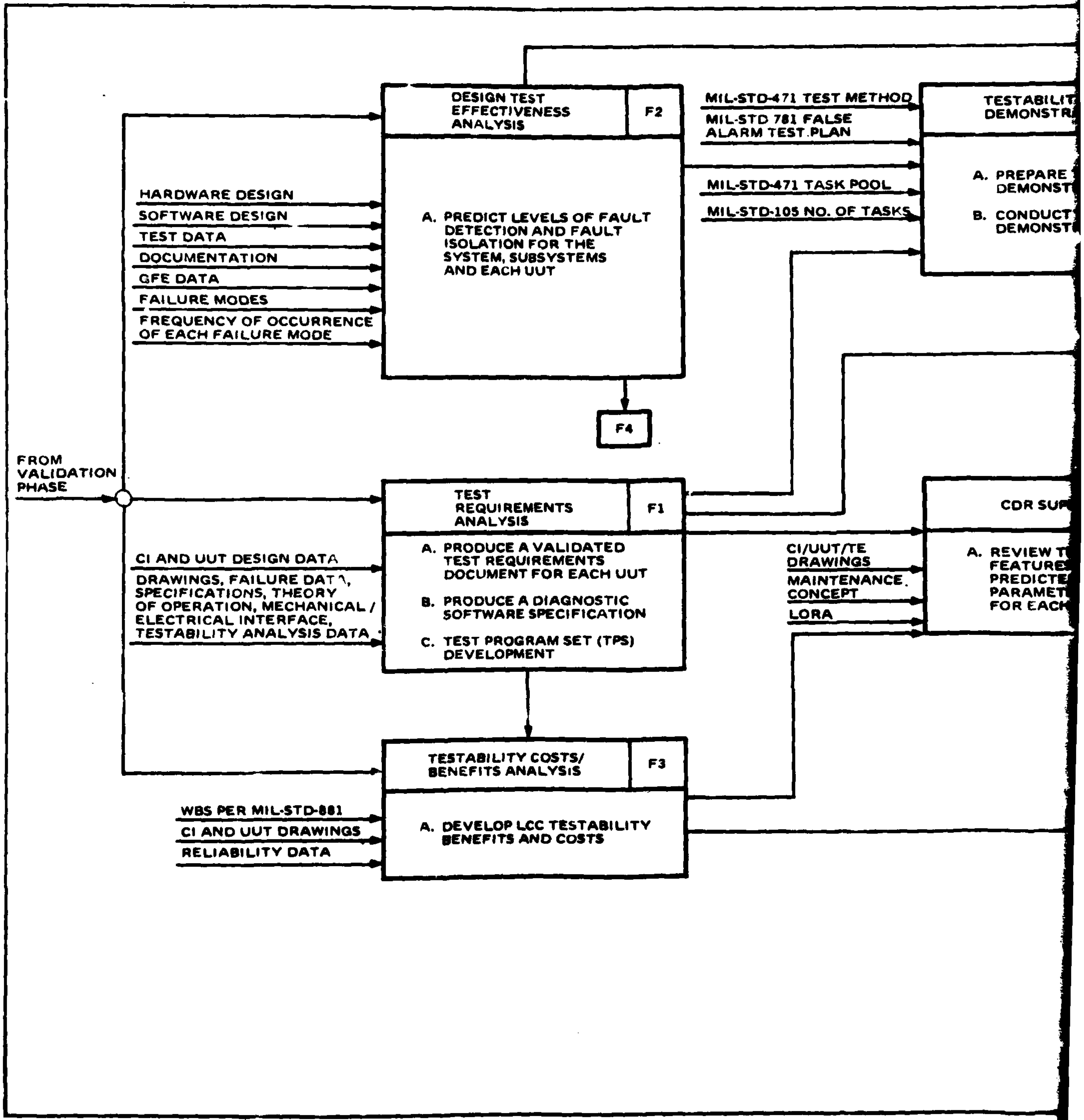
TESTABILITY TASK COMPENDIA

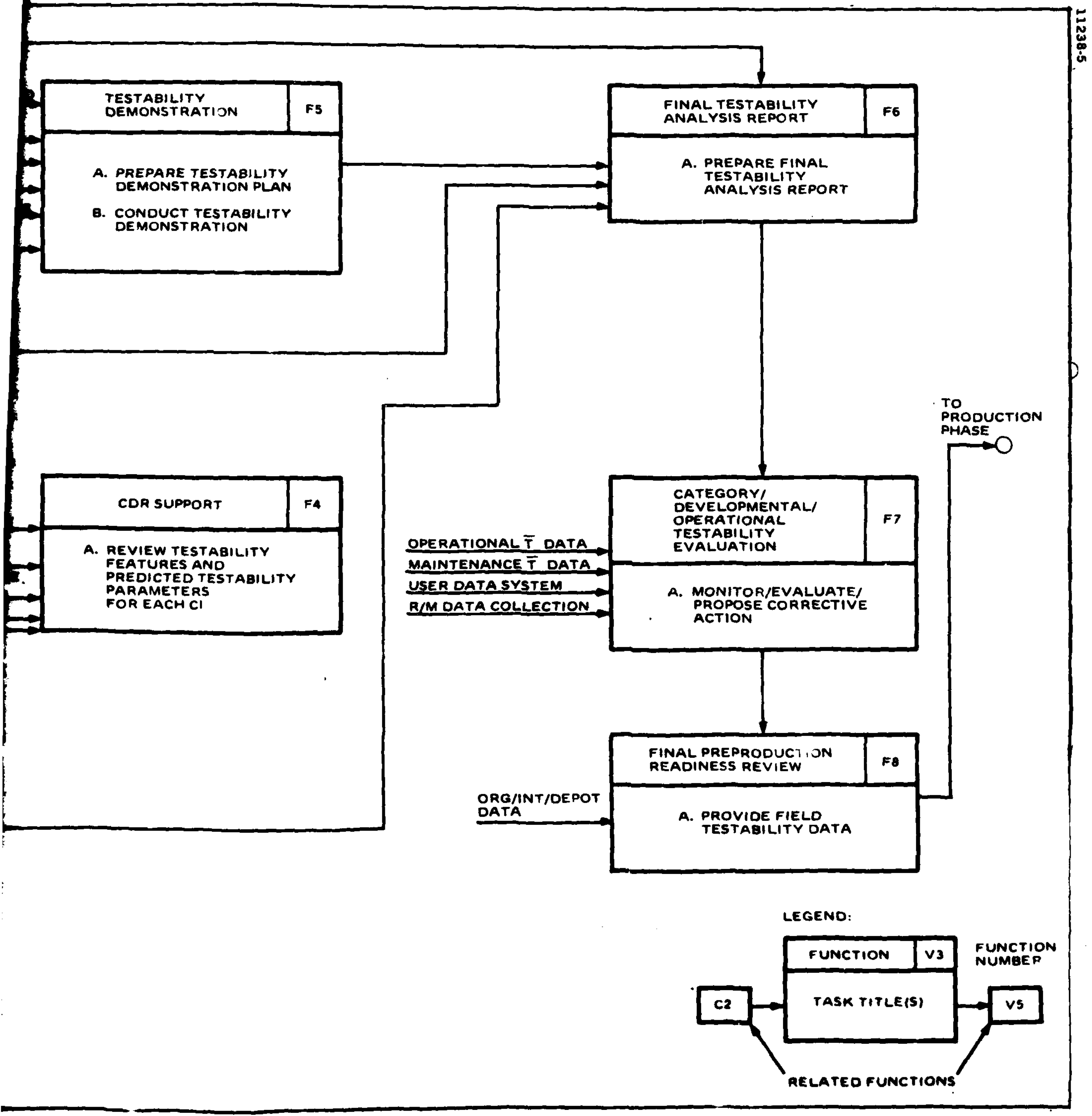
SECTION III - FULL SCALE DEVELOPMENT PHASE

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES


DESIGN TEST EFFECTIVENESS ANALYSIS
F2
A. PREDICT LEVELS OF FAULT DETECTION AND FAULT ISOLATION FOR THE SYSTEM, SUBSYSTEMS AND EACH UUT
HARDWARE DESIGN
SOFTWARE DESIGN
TEST DATA
DOCUMENTATION
GFE DATA
FAILURE MODES
FREQUENCY OF OCCURRENCE OF EACH FAILURE MODE
F4
MIL-STD-471 TEST METHOD
MIL-STD 781 FALSE ALARM TEST PLAN
MIL-STD-471 TASK POOL
MIL-STD-105 NO. OF TASKS
TESTABILIT
DEMONSTR
A. PREPARE
DEMONST
B. CONDUCT
DEMONST
FROM VALIDATION PHASE
TEST REQUIREMENTS ANALYSIS
F1
A. PRODUCE A VALIDATED TEST REQUIREMENTS DOCUMENT FOR EACH UUT
B. PRODUCE A DIAGNOSTIC SOFTWARE SPECIFICATION
C. TEST PROGRAM SET (TPS) DEVELOPMENT
CI AND UUT DESIGN DATA
DRAWINGS, FAILURE DATA, SPECIFICATIONS, THEORY OF OPERATION, MECHANICAL / ELECTRICAL INTERFACE, TESTABILITY ANALYSIS DATA
CDR SU
CI/UUT/TE DRAWINGS
MAINTENANCE CONCEPT
LORA
A. REVIEW T
FEATURE
PREDICTE
PARAMET
FOR EACH
TESTABILITY COSTS/ BENEFITS ANALYSIS
F3
A. DEVELOP LCC TESTABILITY BENEFITS AND COSTS
WBS PER MIL-STD-881
CI AND UUT DRAWINGS
RELIABILITY DATA



Full Scale Development Phase Testability Tasks

2

TESTABILITY TASK COMPENDIUM
Task Reference Number F1A

PHASE: FULL SCALE DEVELOPMENT

FUNCTION: Test Requirements Analysis Performance

TASK TITLE: Produce a validated Test Requirements Document (TRD) for each UUT

TASK OBJECTIVE: The objective of this task is to create and validate a document that serves as a single source of all performance verification and diagnostic procedures and for all equipment requirements to support the UUT in its maintenance environment independent of any specific test apparatus.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering Design Engineers	● Design and Performance	● TRD
● Maintainability Engineering		● $\overline{T}$ Analysis; TRD

COST TRADEOFF INTER-RELATIONSHIPS: Not specifically applicable to this task. The previous testability design effort simplifies the diagnostics and subsequent test program set.

TASK SYNOPSIS:

1. Task Requirements.

1.1 Test Requirements Analysis.

A Test Requirements Analysis defines the functional end-to-end (performance) test requirements and fault isolation test requirements for each item. The input to the analysis process is CI and UUT design data consisting of drawings (schematics, logic diagrams, parts lists, etc.), failure data, performance specification, theory of operation, mechanical/electrical interface definition, and testability analysis data.

1.2 UUT Test Requirements Document.(1)

The Test Requirements Document (TRD) should constitute the formal interface between the activity responsible for detailed hardware design and the activity responsible for Test Program Set development per MIL-STD-1519(USAF). This document serves as a single source of all performance verification and diagnostic procedures, and for all equipment requirements to support the UUT in its maintenance environment, whether supported manually or by ATE or ETE.

The TRD provides detailed configuration identification for UUT design and test requirements data to ensure compatible test programs. The testability analysis performed during the validation phase, refined during the full scale development phase, and documented in the Testability Analysis Report can be used as a partial basis for the Test Requirements Document for each UUT.

1.3 Validation.

The data in the TRD should be validated by actual measurements made on the UUT.

2. Implementation.

2.1 The following summarizes the needs to be addressed by the data assembled in a TRD.(2)

a. Deployed Equipments.

(1) UUT performance verification programs/procedures to provide a screening capability for O or I level maintenance activities.

(2) UUT fault isolation program/procedures to provide a complete diagnostic capability to component level for O, I, or D maintenance levels.

(3) Identification of candidate ATE systems based on the UUT test requirements.

(4) UUT source and test Requirements Supplemental Data to support program/procedures during operational deployment and to provide a capability for:

- (a) Program/procedure debugging and modifications
- (b) Configuration control of support program/procedures
- (c) On-line troubleshooting of UUT and test program problems under deployment conditions

b. Pre-Deployment.

The need for adequate test requirements documentation extends to the preproduction and development phases.

Performance specifications for the prime equipment are being refined during the development process together with the other source data that will be required for the TRD. Examples of such data are schematics, logic diagrams, family tree (configuration), parts lists.

c. Standardization of Data

In order to meet the objectives, the quality and standardization of the TRD must be ensured for the purpose of:

(1) Providing released UUT source data of clearly identified configuration.

(2) Providing clear traceability of UUT test requirements to the UUT source data.

(3) Providing ease of UUT data storage and retrieval through standardized organization of the data.

2.2 Unless otherwise specified, the Test Requirements Document may be prepared in accordance with MIL-STD-1519(USAF).(3) The TRD purpose is to provide the information necessary to test the UUT in the most efficient manner possible and with a minimum of UUT interface while verifying all required performance characteristics.

The TRD may be initially prepared to reflect the preproduction model configuration of the UUT. This version of the TRD is completed when the configuration of the first preproduction model is baselined. The TRD is then revised to reflect the production configuration when the production model is baselined.

Contents of the TRD include the following items:

a. Cover Sheet
b. Approval Sheet
c. Revision Index Sheet
d. Configuration Data
e. General Data
f. UUT Interface Requirements
g. Detailed Performance Characteristics
h. Detailed Test Information
i. Outline Installation Drawings
j. Unit (Main) Assembly Drawings
k. Detail & Subassembly Drawings
l. Wiring Drawings
m. Functional Block Drawings
n. Test Flow Chart

Each of these items is described in detail within the MIL-STD as to the specifics of complete documentation. For the first seven items the appendix to MIL-STD-1519 provides the required format for submittal. By far the bulk of the TRD original work is the detailed test information. Each test to be conducted on the UUT is detailed on one separate test information sheet. The data for each test completely describes all input conditions and the measurements required to perform the test, and the I/O connections specified by connector and pin number.

The TRD is validated by applying the TRD specified I/O to a certified good UUT and verifying that the TRD values are obtained.

2.3 Each test in the TRD should be validated by the contractor. Validation will be accomplished by applying the inputs, loads, etc., specified by the TRD to an acceptable (a certified good) UUT and verifying that the specified values are obtained. A validation certificate can be provided with each TRD. The validation certificate includes the following information: (3)

a. A listing of test numbers, with the actual values obtained from the measurements made during validation testing.

b. Complete listing or identification of:

   (1) TRD
   (2) UUT
   (3) Test equipment
   (4) Test personnel
   (5) Contract number
   (6) Supplier
   (7) Sub-supplier (where applicable)

c. Date testing was accomplished.

d. Signature of test personnel.

e. Signature of the procuring activity representative who witnessed or participated in the testing.

3. Completion Criteria. (3)

3.1 This task is completed upon acceptance of the TRD by the processing activity. Acceptance of the data required by this specification is accomplished by submittal of copy of the validated TRD and validation certificate to the procuring activity. This acceptance, however, is contingent on final review of the delivered materials by the procuring activity. The procuring activity notifies the TRD contractor of final acceptance of the data required by this specification.

TESTABILITY TASK COMPENDIUM
Task Reference Number F1B

PHASE: Full Scale Development

FUNCTION: Test Requirements Analysis

TASK TITLE: Produce a diagnostic software specification

TASK OBJECTIVE: Create a document that serves as a single source of all diagnostic software tests for each UUT.

DESIGN DISCIPLINE TESTABILITY ($\bar{T}$) INTER-RELATIONSHIPS:

$\bar{T}$ INTERFACE	INPUT TO $\bar{T}$	OUTPUT FROM $\bar{T}$
● Application Software		● $\bar{T}$ Analysis

COST TRADEOFF INTER-RELATIONSHIPS: Not relevant to this task.

TASK SYNOPSIS:

1. Task Requirements.

1.1 General Requirements.(4)

The diagnostic software specification describes in detail all the operational and functional requirements necessary to design, test, and maintain the required computer program. In addition, it provides the logical, detailed descriptions of the performance requirements of a digital computer program. The requirements stated in the specification are compatible with all components of the digital system and interfaced systems. However, the specification should not unnecessarily duplicate descriptive material presented in other documents.

1.2 Use may be made of the testability analysis performed during the Validation phase and documented in the Testability Analysis Report as a partial basis for the specification of diagnostic software.(1)

Use may also be made (to the extent available) of the test data developed at the UUT level (for external testing) in designing diagnostic software tests. If needs arise to revise the specification, updated FSD phase data should be incorporated or reflected as appropriate.

2. Implementation.

2.1 The following outline is a normal format for presentation of the material in the Diagnostic Software Specification:(4)

- a. Scope
  - (1) Identification of System and Software Content
  - (2) Functional Summary of System and Software
- b. Applicable Documents
  - (1) Program Definition Documents
  - (2) Inter-Subsystem Specifications
  - (3) Military Specifications
  - (4) Miscellaneous Documents
- c. Requirements
  - (1) Introduction
    - (a) General System Description
    - (b) Peripheral Equipment Identification
    - (c) Interface Identification
  - (2) Functional Description
    - (a) Equipment Descriptions
    - (b) Computer Input/Output Utilization
    - (c) Computer Interface Block Diagram
    - (d) Test Language and Compiler Designation
    - (e) Program Interfacts
    - (f) Functional Description
  - (3) Detailed Functional Requirements
    - (a) Introduction
      - 1. Inputs
      - 2. Processing
        - a. Purpose
        - b. Functional Parameters
        - c. Diagrams of Geometry
      - 3. Outputs
      - 4. Special Requirements
  - (4) Adaptation
  - (5) Capacity

d, Quality Assurance Provisions

(1) Introduction

(a) Computer subprogram testing
(b) Computer program testing
(c) Computer program acceptance testing
(d) System Integration Testing

(2) Test Requirements

(3) Acceptance Test Requirements

2.2 Special attention should be given to computer language used in automatic testing. ATE languages provide the vehicle for expressing, modeling and solving test problems. Bibliography reference number 5 focuses on the nature of different types of test languages, the roles they play in testing and the qualities that make them useful or difficult to implement. The paper discusses:

a. ATE Software Components: support software, control software, test application software and UUT resident software.

b. Test procedures vs. test programs.

c. Special purpose languages

d. Language levels and trends

e. Support software costs

f. Language cost considerations

Certain high-level languages provide better visibility to managers. Natural engineering syntax and vocabulary improve communications between the test programmers and managers.

2.3 The CDRL, DID and this compendium may be used to assess completeness of the testability content of diagnostic software specification.

3. Completion Criteria.

3.1 The task is completed upon government acceptance and approval of the specifications.

TESTABILITY TASK COMPENDIUM
Task Reference Number F1C

PHASE: Full Scale Development

FUNCTION: Test Requirements Analysis

TASK TITLE: Test Program Set (TPS) Development

TASK OBJECTIVE: Assure that each UUT is properly supported by a valid and complete Test Program Set.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● TPS Design Engineering	● ID Description, Drawings	● $\overline{T}$ Requirements, Hardware Review
● TPS Software Engineering	● SW Flow Diagrams, Listing	● $\overline{T}$ SW Requirements, Review
● Program Manager		● $\overline{T}$ Review Comments for TPS

COST TRADEOFF INTER-RELATIONSHIPS: Analysis is required to optimize the range and depth of the test program sets and to determine the TPS approach which minimizes the net costs of TPS acquisition and the costs of manpower and other support resources used in conjunction with the TPS in the deployed phase.

TASK SYNOPSIS:

1. **Task Requirements:**

Every unit to be tested needs an engineered interface with its test device. Modern practice emphasizes the TPS as an interface with Automatic Test Equipment. Some similar form of interface, perhaps lesser scope than a full TPS, is also required if the test device is other than ATE. This task is initiated ahead of critical design review so that the TPS approach may be reviewed there. The TPS themselves are developed later in FSD and fabricated during the production phase.

1.1 Test Program Set.

A TPS is to be prepared for each UUT. A TPS is composed of one test program (TP), one interface device (ID), one Test Program Instruction (TPI), document, and supplementary data. Although an ID may be shared by multiple UUTs, the TP, TPI, and supplementary data are unique to a UUT and its multiple configurations.(6)

1.2 Development and Delivery.

TPS scope includes development, test, quality assurance and configuration management of the family of TPS. Production monitoring is included in Task P1A.

2. Implementation.

MIL-STD-2077(AS)(6) establishes the requirements for the development, test documentation, configuration management, quality assurance, and preparation for delivery of Test Programs (TPs) and that related hardware and documentation to be used in conjunction with an appropriate Automatic Test Equipment (ATE) to test Units Under Test (UUTs).

The following implementation guidance is adapted from MIL-STD-2077 and other sources as referenced. The appendix provides some detailed technical guidance.

2.1 Test Program Set Contents.

The content requirements for the TP, TPI, supplementary data, and ID are as follows:

a. Test Program Content. The TP contains a coded sequence which, when executed by the ATE, provides the system a set of instructions. It consists of:

- Program Heading and Identification
- Self-Test Survey
- Identity Checks
- Safe-to-Turn-On Tests
- Performance Routines (end-to-end test)
- Diagnostic Fault Isolation Routines
- Program Entry Points

b. Test Program Instruction. The contents of the TPI contain that information which cannot be communicated by the ATE under control of the TP (hook-up, probe point locations...) and is required to accomplish testing of the particular UUT such as pretesting data, test data and post-testing data.

c. Supplementary Data. The contents of the supplementary data contain that information necessary to maintain and/or modify the TPS and analyze the TPS and UUT in case of a problem or anomaly during testing. It includes all that additional information essential to a full comprehension of the intent, structure and interrelation of all elements of the TPS.

d. Interface Device. The ID provides the mechanical and electrical connection and signal conditioning, if required, between the ATE and the UUT. It is a requirement to minimize the complexity of IDs subject to the following rules:

(1) Optimize IDs so that as many UUTs as practical can be tested by the same basic ID assembly, with the objective of reducing the total number of IDs required and thus reducing shop storage requirements. The IDs should be designed with a 20% expansion capability. That is, provisions are made in the design of the ID to accommodate unanticipated ID requirements including number of wires, added functions and/or subassemblies 20% greater than the defined requirements.

(2) Each ID will have a minimum Mean Time Between Failure (MTBF) of 1000 hours calculated in accordance with MIL-HDBK-217.

(3) Each ID will be small enough to permit both the ID and any UUT to be physically supported by the intended ATE work surface.

(4) IDs are designed in conformance with the requirements of MIL-T-28800 type III class 4 equipment.

2.2 Test Program Set Development.(7)

The overall process of developing a TPS is shown in Figure 1. The Test Requirements Analysis, Task F1A provides the foundation for TPS development.

a. Test Program Specification Phase. The first task is to develop the Test Program Specification, starting with a functional flowchart (for the tests) and culminating in an English Language Test Document (ELTD) which contains:

- A brief narrative description of all go-chain tests
- A flow chart showing the go-chain and each no-go chain (all stimulus and measurement functions are identified)
- A statement of all ATE operator instructions
- Specific identification of the test adapter

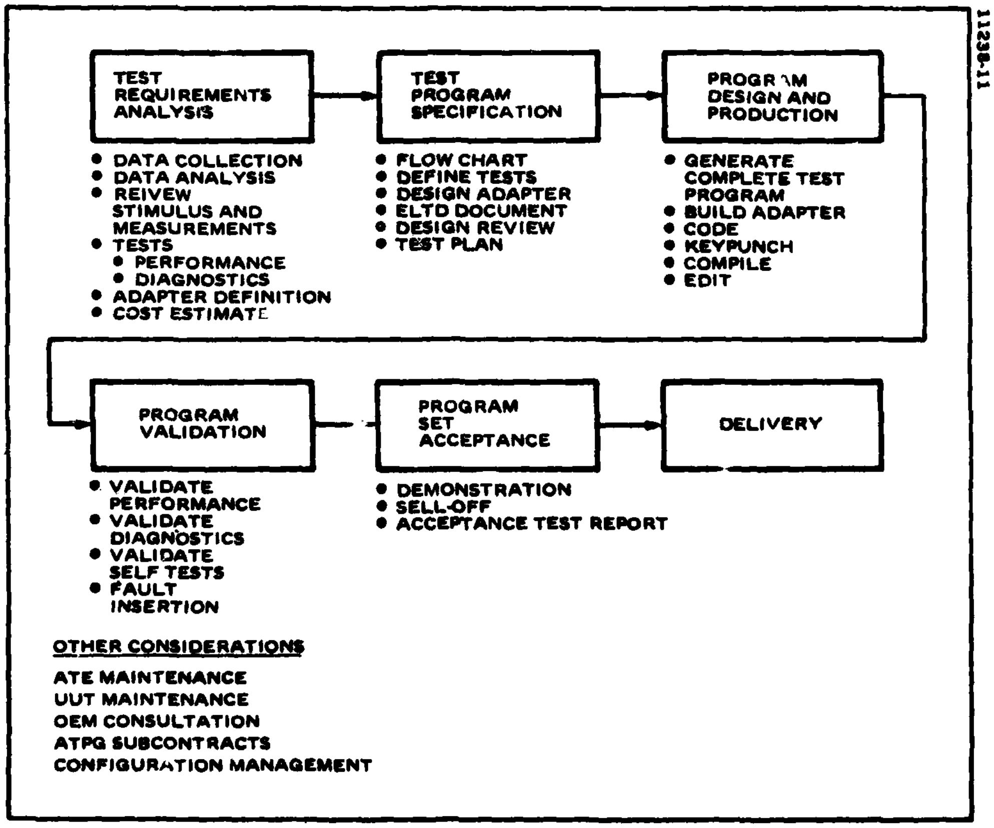


Figure 1. Test Program Set Development

During this specification phase, all stimulus, measurements, and calculations are identified with their appropriate tolerances. All go-chain tests and alignment tests are identified. No-go chains and their respective "condemned" subassemblies or parts are identified.

It is appropriate to conduct a design review to include customer and original equipment manufacturer personnel, with goals as follows:

- Verify the validity of the test approaches
- Answer questions from the TPS designer(s)
- Establish ground rules for selection of UUT faults to be used in validation

b. Program Design and Production Phase. The task is to expand the functional tests into their respective detailed tests, including all stimulus and measurements techniques, to yield a complete test program. At this point, all data required to release the build of the test adapter is known, hence it should be released for build. The detailed tests previously generated are coded into the appropriate high order language statements which are input to the operating system for compilation. This initial compilation is edited and a listing is generated which corresponds to the baseline test program flowchart. The combination of the test flowchart, the test adapter description, and the listing form the initial ELTD to be updated throughout the remaining validation and acceptance phase.

c. Program Validation Phase.

(1) First the test adapters are connected to the ATE and, using a "validation" mode of operation, the test program is executed without the UUT. This is done to verify that stimulus appears at designated interface points, hence provides protection for the UUT. Next the UUT is connected to the ATE. Each go-chain test is executed until certified as correct. Verifications are accomplished on:

- Performance Limits
- Timing
- Operator Instructions
- Adjustment Routines

Test requirements which surface and which were not included in the initial ELTD, but which are obviously necessary, are added as needed during this part of the validation. When the go-chain tests are completely validated and with a good

UUT connected to the ATE, the program is forced through each no-go chain using the "system validation" mode of operation. This is done in order to verify coding, operator interaction, and printouts. Absolute test diagnostic capability is still not validated at this point since the no-go condition was "forced" while the UUT did not in fact harbor a fault.

(2) Next in the validation phase comes fault insertion. While this is strictly an empirical effort, it remains the most important method of assuring a quality program. It is accomplished by inducing faults, one at a time, into the UUT. With each fault, the program is executed to validate the diagnostic capability of the program. Whenever the program fails to detect or correctly isolate a given fault, further analysis is required. In all cases it is necessary to confirm that all selected faults do in fact drive one or more of the UUT's operating parameters beyond specified tolerances.

d. Program Set Acceptance. The Test Program Set acceptance test usually follows a procedure such as this:

- A final ELTD is submitted to the customer in sufficient time for him to review the program and select the faults for acceptance test.
- On acceptance day, the test program is loaded into the ATE in source language and translated by the compiler.
- A good UUT is tested to show that the program recognizes a UUT that functions within specified tolerances.
- Faults, one at a time, are induced into the UUT. These faults are those previously selected by a customer representative.
- If faults are incorrectly isolated, the program is corrected and re-run.
- Upon successful completion, Acceptance Test Records are accumulated and witnessed by all interested parties.

Within 30 (or otherwise specified) days after the acceptance test, the final delivery is made and should include:

- Acceptance Test Report
- Source and Object Programs on magnetic tape
- Final Test Accessory
- Final ELTD

2.3 Configuration Management and Quality Assurance.

a. The configuration of the TPS is managed by documentation and use of procedures for the identification, control, updating and status accounting of TPSs.

(1) Identification may be accomplished by assigning part numbers to each element of the TPS within the contractor's normal part numbering system.

(2) Control and updating are achieved by subjecting all elements of TPS to formal engineering change control.

(3) TPS modifications are subject to management control and are limited to modifications to correct or improve the TPS or to accommodate a UUT change.

(4) Status accounting is used to ascertain the identification of all UUT product configurations, to the design control activity for UUT identification, and to assure that a TPS has been developed for each legitimate product configuration UUT.

b. Quality Assurance. Quality Assurance includes the generation and use of a quality program plan which assures that the procured software will satisfy the support requirements of the prime system.

2.4 Digital Test Program Generation Systems (DTPG).(8)

This guide contains the analysis of 29 viable DTPG systems. By following a three phase selection process the 29 systems are reduced to two or three user applicable systems.

The first phase matches user test programming requirements with the features and capabilities of the 29 simulation systems.

The second phase allows selection based on five areas of technical performance:

(1) IC modeling
(2) Circuit modeling
(3) Good circuit simulation
(4) Fault simulation
(5) Automatic vector generation

The third phase consists of the user exercising the remaining 2 or 3 DTPG systems with a bread board card representative of the various types of circuitry to be encountered. Final choice considers comparison of the test results plus such non-technical aspects as warranty, maintenance agreements and corporate support.

3. Completion Criteria.

3.1 The task of developing the initial test program set (TPS) is complete upon delivery of an acceptable TPS to the customer, following approved validation. Monitoring the production of TPS is included within the requirements of Task P1A. The task of logistics planning would then have the added advantage of knowing the probabilities and problems associated with TPS ambiguity, resulting in more efficient and cost effective maintenance.

APPENDIX F1C1

This appendix consists of adapted excerpts which treat four elements of technique applicable to test program set (TPS) development:

- TPS cost drivers
- Adapter costs
- TPS problems and solutions
- Automatic Test Program Generation

1. Test Program Set Cost Drivers

a. Testability (9)

The largest cost driver in the development of a Test Program Set (TPS) is the testability* of the unit to be tested. Testability affects the analysis, interface adapter design, integration and debug cost as well as the life cycle maintenance costs of the UUT.

The factors of testability that impact TPS costs include the following:

- Test Point Design and Placement
- UUT Initialization
- UUT Accessibility
- UUT Packaging
- Adjustments and Select-at-Test Components

b. TPS Isolation Ambiguity versus UUT Sparing (9)

The logistics sparing for UUTs normally takes place before the TPS is designed, and is based on statistical failure rates of components or modules. When the test program cannot reach unambiguous isolation of UUT failures, the maintenance technician is often faced with a lack of spares. Experimentation, substitution, and repair by stages will usually result in the UUT being restored to service, but the costs involved are considerable. The test station time required to repair such a failure often results in the backup of other failed UUTs in the maintenance pipeline creating a maintenance backlog.

It is suggested that sparing be accomplished only after the test program analysis is completed.**

---

* "Testability" here refers to the ease and simplicity of test inherently contained in the UUT. -Ed.

**Alternatively, the quantitative spares analysis should take ambiguities into account. -Ed.

The task of logistics planning would then have the added advantage of knowing the probabilities and problems associated with TPS ambiguity, resulting in more efficient and cost effective maintenance.

c. Program Structure (9)

One of the reasons for high TPS maintenance costs is the total lack of guidelines and specifications regarding TPS structure. The classic problem with all software maintenance is that the original programmer is not available for the correction of the error, and a new individual must try to figure out what was meant or intended before a change or correction is attempted.*

Well-defined and meaningful entry points and program documentation can help reduce field maintenance time, and also reduce the time required to identify and correct test program defects. Entry point tables that identify the function being tested reduce repair verification time and also help in establishing program structure.

d. Other Significant Cost Contributors (7)

Other significant cost contributors not readily visible in the TPS development process are:

- ATE Maintenance
- UUT Maintenance
- Original Equipment Manufacturer (OEM) consultation
- Automatic Test Program Generator (ATPG) subcontracts
- Configuration Management

2. Multi-purpose Adapters for Cost-reduction (10)**

Assume a system of 100 equipments and 2000 modules with sets of adapters required for 20 sites at $200 per adapter. The recurring costs of adapters can vary as follows:

- If each module requires a unique adapter(2000 adapters x 20 sites x $200/adapter) = $8,000,000
- If an average of five modules are served by each unique adapter (2000 ÷ 5 adapters x 20 sites x $200/adapter) = $1,600,000

---

* This problem is common to all programming, yet it can be avoided by simple standardization of and adequate development of programming documentation. - Ed.

** This excerpt is presented to illustrate a potential tradeoff. The source article did not treat all recurring and non-recurring costs impacted by the adapter decision. Numbers are quoted directly from the source. Intuitionally, the more complex the use of an adapter, the higher its cost may be. Also, there are impacts on the non-recurring engineering. -Ed.

- If each equipment is served by a unique adapter (100 equipments x 20 sites x $200/adapter) = $400,000

The analysis needs to be extended to consideration of non-recurring as well as recurring costs and must also consider operating and support cost impacts.

3. TPS Problems and Solutions[11]

Inaccuracy of diagnosis after the Test Program Sets are delivered to the user has been troublesome. (In a survey):

- Undetected non-UUT (ATE plus interconnection) related failures caused 60% of the mis-diagnosis.
- Out of scope conditions were responsible for some 30% of diagnostic problems (multiple-failures, chassis wiring, non-standard failure modes, etc.)
- Only 8% of the erroneous fault isolation could be traced to UUT oriented TPS software errors.

Conservative estimates indicate that the customer loses the equivalent of more than 30% of the original procurement cost per year due to this type of mis-diagnosis. The cost of corrective action is estimated at less than one-tenth this amount if incorporated before TPS production is started and is significantly less if done concurrently with TPS development.

a. Candidate Corrective Actions.

(1) Improve ATE real-time self-test capability.

- Utilize local microprocessor monitoring of test equipment functions.
- Implement hardware design which allows optimum failure monitoring in real-time.
- Provide system software modules to allow real-time monitoring during testing.

(2) Improve ATE testability.

- Provide necessary hardware to allow extensive self-test of the ATE.
- Generate ATE resident software for self-test.
- Allow running of subsets of self-test software on an as-required basis.

(3) Include in-process testing of non-UUT resources prior to use.

- Provide ATE self-test as part of UUT performance testing.
- Provide stimulus and measurement limits to suit specific UUT tests.

(4) Include in-process calibration to attain required stimulus and measurement accuracies.

- Provide measure-after-apply techniques for stimuli.
- Provide stimulus standards in ATE for calibration of measurement devices.

(5) Provide external and internal hardware to achieve near 100% ATE confidence at the UUT connectors.

- Use additional hardware in interconnection device.
- Apply external wraparound techniques.

(6) Expand data collection and use in developing meaningful models for ATE effectiveness and TPS quality to include*

- Failure data
- TPS development times
- On-station activity.

4. Automatic Test Program Generation (ATPG).(12)

Digital Automatic Test Program Generation is defined as a computer and/or computer program which aids in the generation of test programs for digital electronic assemblies in an automated manner. It includes both simulators and simulators with automatic stimulus generation and is used for both the development and maintenance of digital test programs. It exists because the manual generation of digital test programs is complex, expensive, skilled-labor-intensive and error prone. The figure shows the general ATPG block diagram. The electronic assembly schematic information is fed into the system. The ATPG system draws on an IC component library to model the schematic. A check of the model is accomplished. The stimulus is then generated, either manually or automatically. The ATPG system then simulates the electronic assembly and determines the fault detection and fault isolation percentages. The patterns are then translated into the specific ATE language desired. The final output is a Unit Under Test (UUT) test program.

---

* Data as appropriate should be provided to vendors for product improvement.

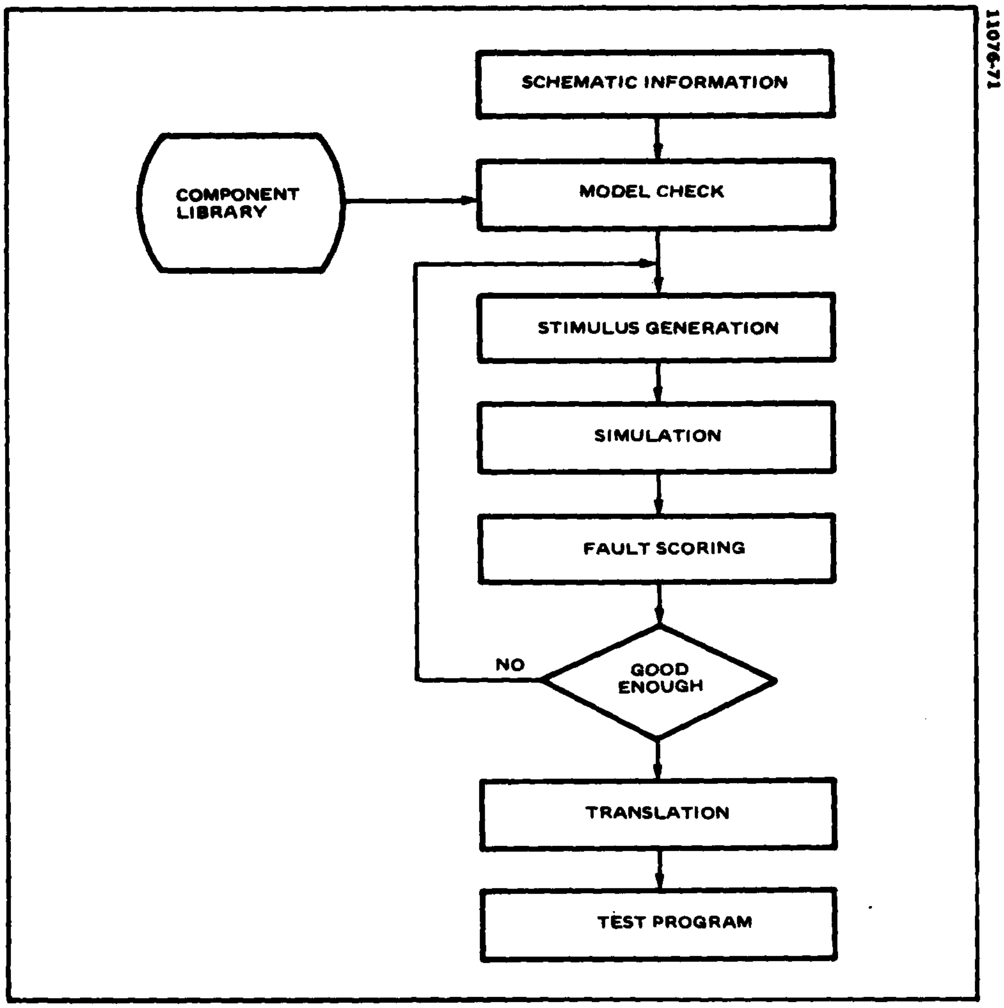


Figure 2. General ATPG Block Diagram

The generation of test programs is proportional in complexity to the complexity of the electronic device which is to be tested. The more complex the device, the more difficult it is to generate a test program for that device. Testability plays a key role in the generation of test programs. Good testability can greatly ease test program generation and test program maintenance.

A choice of ATPG systems is available for use. The following criteria are suggested for use in selection of a system.

- Ability to Model ICs
- Stimulus Generation Capability
- General Ease of Use
- ATE Compatibility
- System Maturity
- System Cost - Acquisition and Use
- Vendor Stability
- Growth Capability
- System Maintenance Availability

TESTABILITY TASK COMPENDIUM

Task Reference Number F2A

PHASE: FULL SCALE DEVELOPMENT

FUNCTION: Analysis of Design Test Effectiveness

TASK TITLE: Predict the levels of fault detection and fault isolation for the system, subsystems and each UUT.

TASK OBJECTIVE: Achieve a detail design which allows for optimal detection and isolation of failures and minimizes the occurrence of undetected failures.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● System Engineering	● Partitioning, System Level BIT/ETE, System Functional Diagram	● $\overline{T}$ Analysis Results (detail design recommendations)
● Design Engineering	● UUT Test Points, HW BIT, Inherent $\overline{T}$ Features Schematics, Logic Diagrams	● $\overline{T}$ Analysis Results (detail design recommendations)
● Application Software	● SW BIT, Test Program Flow Charts and Program Lists	● $\overline{T}$ Analysis Results (detail design recommendations)
● Reliability Engineering	● FMEA, System/Module Failure Rates	● $\overline{T}$ Analysis Results (detail design recommendations)

COST TRADEOFF INTER-RELATIONSHIPS: As the design evolves, consideration given to the assurance of observability and controllability may add to design costs. These increases will be offset by reduced costs in test program set development, by reduced costs in conducting test in operations and by reduced costs in logistics (manpower, spares, and test equipment usage).

TASK SYNOPSIS:

1. Task Requirements.

1.1 A formal analysis of failure modes of the final design is required as a followup to the design interfacing conducted in Function V2, Incorporation of Testability into System Design and Function V3, Performance of Test Method Tradeoffs and subsequent coordination efforts.

The task is initiated in conjunction with preparation for the Critical Design Review and continues until the test effectiveness can actually be demonstrated by evaluation of developed system hardware.

The requirement is to determine that expected failures can be detected and isolated so that mission performance effectiveness is achieved and that the minimum acceptable number of faults go undetected. Measuring failure observability and test controllability is a primary means of assessing testability effectiveness in this regard.

2. Implementation.

The total task consists of two parts, a hardware failure analysis to analyze test effectiveness and a testability analysis model to analyze the inherent observability/controllability of the configuration. Maximum use should be made of previous analysis tasks (Function V4) performed during the validation phase. The analyses are used by the contractor to support pre-baseline design changes and to provide a vehicle for government review at both system and subsystem CDRs. Portions of this task are closely related to Test Program Set development of Task F1C.

The overall concept of this task is to analyze the failure modes in order to obtain measures of observability and controllability. The analyses should be computerized as feasible. The appendix provides some primitives for use in analysis.

2.1 General Notes for Detailed Analysis.(1)

The failure population is the basis for test derivation (BIT and/or external test) and the basis for test effectiveness evaluation. With respect to the failure modes, an initial step is to determine hardware partitions for failure effects analysis considering accuracy required, cost of test generation and simulation, and standardization and commonality.

a. Testability building block. The lowest level of hardware partition may be referred to as a Testability Building Block (TBB). A TBB may or may not correspond to a physical partition of the system but typically represents a LSI component or a small printed circuit board. Component structures and interconnections may also be fitted into TBBs such that the relevant failure population may be accurately modeled.

b. "Model" validation.* The structure of each TBB should be verified. A simulation technique is needed to apply appropriate portions of the functional end-to-end test and compare simulated responses with predicted responses (or with responses obtained from a known good hardware unit, in subsequent tasks).

*Subtasks 2.1b and 2.1c may be conducted as part of TPS validation at such time that the TPS is available.

c. Test stimulus generation.* This derives test sequences for each TBB using the most cost-effective methods. Maximum use should be made of functional test sequences developed for end-to-end testing. Test algorithms should be derived so as to facilitate BIT and tester implementations considering software looping and memory constraints.

d. Failure response data. For each specified failure in the TBB, the output response to the input test stimulus should be determined through a simulation technique. Using appropriate failure detection criteria, a record should be kept of whether or not the failure was detected, and if detected, of the expected (good) output signature, together with the predicted output signature of the failed unit.

e. Undetected failures. Undetected failures should be examined to determine if:

(1) The failure is not detectable by any test sequence. Any failures which are impossible to detect due to redundancy should be considered non-relevant to the analysis.

(2) The failure is potentially detectable, but the test sequence is deficient.

(3) The failure is potentially detectable, but the unit's hardware design precludes the use of test sequence of reasonable length or complexity.

f. False alarm rate. All GO paths of BIT software and Test Program Sets should be validated using a known good system or UUT prior to test design approval. The correct operation of concurrent hardware BIT and application software error processing routines should be verified by analysis of the design. For analog systems, or digital systems without software, all GO paths of each BITE indication should be validated using a known good system or UUT prior to final hardware design acceptance.

g. Failure detection times. Failure detection time (the time which elapses between the occurrence of a failure and the detection of the failure by the test process) should be expressed as two or more time intervals (representing the detection latency of generic test approaches, e.g., concurrent BIT, periodic BIT), and the proportion of failures falling within each interval. For example:

Maximum Detection Time	% Failures
Less than 1 second	25
Between 1 second and 10 seconds.	65
More than 10 seconds	10

h. Failure isolation times. The average (or maximum) time to isolate failures should be predicted using the average (or maximum) length of

---

*Subtasks 2.1b and 2.1c may be conducted as part of TPS validation at such time that the TPS is available.

the diagnostic test sequence plus an estimation of time for any manual intervention required. Times should be predicted for both BIT and ATE.

i. Government furnished equipment data. Testability parameter values should be requested from the procuring agency and used in the testability prediction. If the values are unavailable or unknown, estimate the values. If the estimated or furnished values are incompatible with the intended use, or analysis indicates that the system/equipment will not satisfy the operational or maintainability requirements based upon these values, these problem areas should be identified and the procuring activity advised with proposed alternative courses of action.

j. Corrective action. This may include additional patterns in the test sequence, where feasible, to meet failure coverage and isolation requirements. If additions to the test sequence are not possible or are not reasonable, the T engineer can propose appropriate design changes for the hardware to improve its controllability/observability characteristics. The proposed modification should be modeled and simulated using modified test sequences to determine if the changes improve the failure coverage and isolation sufficiently.

2.2 Observability/Controllability Analysis.(1)

A Testability Analysis for each Configuration Item (i.e., potential UUT) should be developed and maintained. Maximum use should be made of analysis developed during preliminary design to determine inherent observability and controllability. The overall testing structure should be represented by a hierarchy of analysis representing various levels of testing. Higher levels in the hierarchy should be defined in terms of the lower level blocks and the stimulus/response requirements for each lower level block. An analysis may correspond to any physical or functional grouping. Certain analyses represent the testing structure for UUTs and thus support TPS development. Each level of Testability Analysis should permit the following determinations:

a. Identify the response observation points (for BIT, ATE, and manual test for this level.

b. Identify the test modes and stimulus injection points for this level.

c. Verify that signal paths and control paths exist to provide stimulus to lower level blocks as determined by their test requirements and identify the stimulus required.

d. Verify that signal paths and control paths exist to propagate the resulting test response of lower level blocks to observation points at this level and identify the resulting failure responses.

2.3 Military Documents.

The following documents may aid in task performance:

a. Failure Population Definition, MIL-STD-471, Maintainability Verification/Demonstration/Evaluation

b. Failure Mode and Effects Analysis, MIL-STD-2070(AS), Procedures for Performing a Failure Model, Effects and Criticality Analysis for Aeronautical Equipment

3. Completion Criteria.

This task should be merged with the development of Test Program Sets and the Testability Demonstration. It is complete when the Testability Demonstration results have been approved by the government.

APPENDIX F2A1

This Appendix presents details of two measures which may aid in a determination of effectiveness: (1) Weighted Failure Coverage, and (2) Failure Resolution.[1]

1. Failure Coverage.

The unweighted failure coverage is defined as K/S where K is the number of failures detected and S is the total number of failures in the failure population, corrected for impossible detects. If the analysis data base includes failure rate data, the weighted failure coverage is calculated as the sum of the failure rates of the detected failures divided by the sum of the failure rates of all the specified failures. Individual failures within a component should be assigned an equal proportion of the component's total failure rate unless more accurate failure data are available and feasible to apply.

2. Failure Resolution.

The degree of failure isolation is calculated using the following methodology.

a. Failure signature data. Data are required which correlates each detected failure with the signature it produces during testing. The data are most conveniently ordered by signature and by failing module within each signature (fault dictionary format).

b. Substitution method. The failure resolution calculations depend upon the substitution method used to effect repairs. If all modules under a signature are replaced as a block, equation (3a) is used. If one module of the signature group is replaced and the test rerun for PASS/FAIL, equation (3b) is used (both equations are stated under Paragraph 2d, Calculation, below).

c. Notation.

N = number of unique signatures in dictionary

i = signature index

M_i = number of modules listed in signature i

j = module index within signature

F_{ij} = number of faults in module j which produce signature i

k = failure index within module

λ_{ijk} = failure rate for kth failure within jth module within ith signature (see note)

$\lambda_{ij} = \sum_{k=1}^{F_{ij}} \lambda_{ijk}$ = failure rate for jth module for failures providing signature i (see note)*

$\lambda_{i} = \sum_{j=1}^{M_i} \lambda_{ij}$ = failure rate for failures producing signature i

$\lambda = \sum_{i=1}^{N} \lambda_{i}$ = overall failure rate

Mmax = maximum (M_i) = worst case isolation

* NOTE: If detailed failure data is unknown, apportion module faults as:

$$\lambda_{ij} = \left(\frac{F_{ij}}{\text{Total Module Faults}}\right) \lambda_{\text{module}}$$

d. <u>*Calculations*</u>.

A_ℓ = % signatures which have an ambiguity of ℓ modules

$$= \frac{1}{N} \sum_{i=1}^{N} (X_i) \times 100 \qquad \text{where } Xi = \begin{cases} 1 \text{ if } M_i = \ell \\ 0 \text{ if } M_i = \ell \end{cases} \tag{1}$$

A'_L = % signatures with ambiguity $\leq L$

$$= \sum_{\ell=1}^{L} A_\ell \quad \text{where one or more values of L are defined in the specification.} \tag{2}$$

RR_i = replacement rate for signature i based upon failure rates

"Replace all" strategy:

$$RR_i = \lambda_i \tag{3a}$$

"Replace one at a time and retest" strategy:

$$RR_i = \frac{1}{M_i} \sum_{j=1}^{M_i} j\lambda_{ij} \tag{3b}$$

PS_i = % replacements due to signature i = $\frac{RR_i}{\lambda} \times 100$ (4)

PR_ℓ = % replacements due to signatures containing ℓ modules = $\sum_{i=1}^{N} XiPSi$ (5)

PR'_L = % replacements with ambiguity $\leq L = \sum_{\ell=1}^{L} PR_\ell$ (6)

TESTABILITY TASK COMPENDIUM
Task Reference Number F3A

PHASE: FULL SCALE DEVELOPMENT

FUNCTION: Testability Cost/Benefits Analysis

TASK TITLE: Develop Testability Benefits and Costs

TASK OBJECTIVE: Provide visibility for non-recurring and recurring costs, penalties and offsetting benefits and savings due to incorporation and exploitation of testability in the prime system.

DESIGN DISCIPLINE TESTIBILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● Application Software	● Differences in Number of Instructions, Memory Size, Documentation	● Guidance to all other disciplines to assist in developing valid information for Cost/Benefits Analysis. Guidance inincludes description of baselines for comparisons and differences associated with Testability.
● Design Engineering	● Differences in Development (Design) Costs, Number Circuits, Documentation.	
● Logistic Support	● Differences in Spares, Repair Turnaround Times.	
● Maintainability Engineering	● Differences in MTTR	
● Maintenance Engineering	● Differences in Repair Levels, Facilities, Skill Levels	
● Support and Test Equipment	● Differences in Test Equipment Complexity, Cost, Maintenance	
● Manufacturing Test Engineering	● Differences in Fault Detection Time, Fault Isolation Time, Alignment/Calibration Time, Checkout/Selloff Time	
● Training	● Differences in Personnel Training Time, Complexity of Training Preparation, Materials	
● Maintenance Documentation	● Differences in Manual Preparation, Complexity	
● Reliability	● Differences in Failure Rates	
● Life Cycle Costing	● Differences in LCC & LCC elements; Guidance in Use of Current LCC Techniques	● Documents both tangible & intangible Testability Impacts & provide inputs to LCC analysis.

COST TRADEOFF INTER-RELATIONSHIPS: The task itself treats cost differences extensively although it represents a recapitulation rather than a cost-driving effort.

TASK SYNOPSIS:

1. Task Requirements.

The essential requirement is to achieve a presentation of benefit vs. penalties which is clear, logical and auditable. In presenting costs, the differences (between design approaches reflecting varying degrees of testability) need to be emphasized to a greater degree than the absolute costs.

2. Implementation.

The Design Discipline Interrelationship chart at the heading of this task compendium is indicative of the implementation. The LCC analyst is probably the most valuable of the contacts for this task, inasmuch as the techniques to be applied are primarily those of LCC analysis. The techniques include those of various repair level analysis methods, (e.g., AFSC 800-4, MIL-STD-1390) as well as USAF Logistic Support Cost and LCC models. Beyond these techniques, each testability analyst will need to tailor methodology and analysis for application to each particular program. One literature extract is quoted which provides some detailed guidance in analysis of Test Program Set costs and benefits.

2.1 Costs Analysis.

Costs fall primarily into development and recurring areas.

a. Development costs. The costs of a development program which are attributed to testability design are visible and identifiable through the assignment of appropriate work elements in the Work Breakdown Structure. Costs for testability design, analysis, and data preparation during a formal testability program should be included.

b. Recurring costs and penalties. Identify those costs and operational penalties associated with the incorporation of testability into the system or equipment. Itemized testability costs should include, but may not be limited to, the following:

- Per unit cost of additional hardware required for BIT and testability capabilities
- Volume and weight required for additional hardware, additional connectors, and increased modularity
- Power required for additional hardware
- Computer memory required for test programs
- Possibility of system interruption due to failure within BIT circuitry
- Reliability impact due to additional hardware

c. Test Program Set Development Costs.(13) The development of test program sets (TPS) for the prime equipment is an initial nonrecurring cost. TPS development costs must be considered for both ATE and MTE. It is important to retain visibility of costs incurred independently of a concerted testability program; most procurements require many of these elements even without mention of optimized testability. Test Program Set Development includes the following elements of cost:

	Element	Labor	Material
(1)	Acquire basic data on UUT for test analysis	In-house development or detailed review of vendor data	Outside purchase from vendor
(2)	Develop test strategy and document in the form of Diagnostic Flow Charts (DFC) and test setup diagrams compatible with tester to be used.	Perform test requirement analysis (TRA). Prepare test requirement document (TRD), including test interconnect diagram.	Digital stimulus/response data may be purchased from vendor or produced using automatic test program generation techniques.
(3)	Interface hardware design and model	Document interface device (ID) hardware and build development model.	Hardware costs for material & electrical components.
(4)	Test Program Instructions (TPI)	Develop step-by-step procedure for on-station testing of UUT including operator actions to correct malfunctions detected.	Artwork costs & printing

	Element	Labor	Material
(5)	Code/Compile (ATE) or Test Procedure (MTE)	Generate test program software using tester's higher order language (ATE) or prepare detailed test procedure (MTE).	Compiler operations and maintenance (ATE)
(6)	TPS Integration	Verification of test programs integrity by actual demonstration on station a) UUT test performance debug b) UUT test diagnostic debug c) Fault simulation d) Test Program, ID, and TPI Updates	Repair facility support and tester maintenance
(7)	Formal Sell-Off (Validation)	Demonstration for inspection personnel and customer of TPS integrity. Includes functional testing & sample fault insertion.	Same as (4), (5) and (6)
(8)	Verification	Demonstration of first production article on tester and fleet introduction.	Field service expenses

The complexity of TPS design will cause a wide variance in total manhours due to degree of testing required for the following reasons:

(1)	Level of Repair	Functional test with no diagnostics to full diagnostics.
(2)	Tester compatibility	Complexity of ID design to mate tester to UUT.
(3)	Test ambiguity	Degree of fault isolation to groups of SRUs, single SRUs or group of components.
(4)	Software update capability (ATE)	The flexibility of ATE software update from on-station patching (simple) to batch compilation on a separate computer (complex).

(5)	Verification and Validation (V&V) Requirements.	Degree of sampling required to sell off TPS.

2.2 Fundamental Benefit Areas.

Cost elements treated in current LCC techniques should be screened for applicability to this task. Penalties and benefits may be expressible in financial terms but will generate from analysis of performance, operational and support impacts. The following excerpts(1) provide a point of departure for defining detail elements to be treated.

a. Development benefits. Estimate the cost reductions (due to the testability program) in:

- Test Program Set (including interface device) size, complexity, development time and cost.
- Diagnostic software size, complexity, development time and cost.
- Automatic test generation software complexity, effectiveness and cost.
- Factory test time and cost for all levels of assembly.
- TPS/diagnostic software costs for later hardware modifications.

b. Operational and support benefits. Estimate the cost benefits of the testability program on system readiness in terms of fewer systems assets needed to meet operational readiness. Support costs are substantially reduced particularly in the following elements:

- Technical Manual acquisition
- Initial and replacement training
- Initial and replenishment spares
- Skill levels
- Test and repair manhours
- Test equipment maintenance
- Transportation of repair items

2.3 Documentation.

The documentation of the results of the cost/benefits analysis should show that inputs from all testability interfaces are complete and valid and all testability affected items are accounted for.

3. Completion Criteria.

The task is completed upon government acceptance and approval of the documented results.

TESTABILITY TASK COMPENDIUM
Task Reference Number F4A

PHASE: FULL SCALE DEVELOPMENT

FUNCTION: CDR Support

TASK TITLE: Review testability features and predicted testability parameters for each CI

TASK OBJECTIVE: Assure that testability features and predicted testability parameters are included as part of the design approved for development into the production configuration.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACES	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● All Design Activities	● Consultation and Coordination	● Consultation and Coordination
● Program Management	● Critique of CDR Review Material	● CDR Review Material, Presentation to Contractor/Government Program Management

COST TRADEOFF INTER-RELATIONSHIPS: Not relevant to this task.

TASK SYNOPSIS:

1. **Task Requirements.**

The task requirement is to present at the CDR, the testability aspects of the detailed engineering design so that they are incorporated as requirements to be met during evolution of the design into manufacturing configuration. Proper testability representation in the CDR process obviates the need for subsequent complex and expensive redesign to accommodate testability.

2. **Implementation.**

Extensive preparation in terms of liaison with other disciplines is advisable so that the Testability position presented at CDR will be non-controversial. CDR should be followed up by monitoring actions assigned which are relevant to testability.

2.1 Preparation.

Preparation for the CDR consists of review of all generated testability materials and review of the development specifications, followed by organization of the major testability elements for concise, unambiguous presentation. Coordination should be conducted as necessary to assure validity of data and understanding by the other disciplines of the testability viewpoint.

The source documents and data include validation phase testability analysis report, maintainability documentation, detailed engineering drawings of each CI, UUT, Test Equipment, diagnostic software design, flow diagrams, and backup material.

2.2 CI Testability Data.(1)

Testability data for each CI is reviewed, including built-in test methods (hardware, firmware and software) and predictions on failure coverage, failure isolation levels and times, and false alarm rates. The achievement of qualitative testability requirements is verified as part of the Functional Configuration Audit. The detailed engineering drawings of the product specification are reviewed and prototype hardware is inspected to insure compliance with test requirements in the development specification, including:

a. Mechanical and electrical compatibility between the prime equipment and on-line readiness monitoring equipment for all parameters specified to be monitored.

b. Modular construction of the equipment to support testing and maintenance requirements.

2.3 UUT Testability Data.(1)

Testability data for each module of the CI is reviewed, including test interfaces, test control, and test access for both ETE and BIT environments. Predictions are made on the levels of failure coverage and failure isolation (within the module) that may be achieved. The detailed engineering drawings of the product specification are reviewed and the prototype hardware is inspected to insure compliance with test requirements in the development specifications including the mechanical and electrical compatibility between the specified ETE and each prime equipment module designated to be a unit under test for that ETE.

2.4 Test Consistency.(1)

The capabilities of built-in-test, external test, and the corresponding maintenance documentation for each are reviewed for consistency with the maintenance concept and Level of Repair Analysis, plans for orderly progression from organizational level testing to intermediate/depot level testing are reviewed. The contractor will plan for maximum utilization of designated ETE during production and test and evaluation.

2.5 Configuration Control.(1)

The configuration control to be exercised in accordance with MIL-STD-480 over each CI, diagnostic software CPCI, ETE, and Test Program Set is reviewed.

3. Completion Criteria.

3.1 Government Approval.

Government approval of the equipment/development specifications, together with approval of CDR minutes comprise completion of the task. Actions arising from the CDR should be assigned as management attention items.

TESTABILITY TASK COMPENDIUM
Task Reference Number F5A

PHASE: FULL SCALE DEVELOPMENT

FUNCTION: Testability Demonstration Plan Preparation

TASK TITLE: Prepare Testability Demonstration Plan

TASK OBJECTIVE: Document for government approval the descriptions of the procedures for conducting the Testability Demonstration.

DESIGN DISCIPLINE TESTABILITY ($\bar{T}$) INTER-RELATIONSHIPS:

$\bar{T}$ INTERFACE	INPUT TO $\bar{T}$	OUTPUT FROM $\bar{T}$
● Design Engineering	● HW BIT Features, Equipment $\bar{T}$ Features Designed In	● $\bar{T}$ Demonstration Plan Requirements
● Application Software	● SW BIT Features, SW Diagnostic Tests	● $\bar{T}$ Demonstration Plan Requirements
● Support Equipment	● ATE/Manual Test Equipment Availability	● $\bar{T}$ Demonstration Plan Requirements
● Test Engineering	● Test Program Features	● $\bar{T}$ Demonstration Plan Requirements
● Program Manager	● Test Schedule & Review Dates	● $\bar{T}$ Demonstration Plan

COST TRADEOFF INTER-RELATIONSHIPS: Not applicable to this task.

TASK SYNOPSIS:

1. Task Requirements.

This task requires the preparation of a plan, formatted in accordance with the CDRL for the conduct of a testability demonstration. The demonstration may also be governed to some extent by the quality assurance procedures of the contract. The demonstration is to be performed to verify the achievement of specified testability parameters on major end items, similarly to the demonstrations of achievement of maintainability parameters conducted in accordance with MIL-STD-471.

This task consists of development of the plan by which the actual demonstration will be conducted. Task Reference Number F5B provides for actual conduct of the testability demonstration.

2. Implementation.

This task involves preparation of the testability demonstration plan. A testability demonstration evaluates the effectiveness of on-line tests and off-line test interfaces.

The techniques which follow here address the concept of testability demonstration, the testability task pool and demonstration procedures as background to understanding plan formulation. A candidate DID from the literature is also shown.

2.1 Testability Demonstration.

Demonstration tests are performed on major end items, including complete systems, using the anticipated test environment (test equipment, test software, and test documentation). The testability demonstration should be accomplished during Full Scale Development qualification testing utilizing contractor personnel and contractor facilities, and should be monitored by the procuring activity.(1)

Demonstration procedures include the introduction of actual failures into equipment for the verification of BIT, test software, and maintenance error dictionaries. This demonstration is in addition to any specified Maintainability Demonstrations which address maintenance procedures, equipment accessibility, technician skill levels, etc., as well as fault detection and isolation considerations.(14)

a. Testability Task Pool Concept. The demonstration of testability parameters should be accomplished through the completion of several testing tasks. These tasks are selected at random from a task pool defined by the contractor prior to the demonstration. Each task defines one failure, the method of inserting or simulating the failure, the expected failure response, and the detailed operating procedures for obtaining the initial failure detection indication. Failures may be simulated by introduction of faulty parts, deliberate misalignment, open leads, shorted parts, etc. The task pool should contain at least twice the maximum number of tasks required for demonstration and should be specified in the Testability Demonstration Plan. The task pool should be prepared by the contractor in accordance with the stratification procedure described in Appendix A of MIL-STD-471 to insure that a proportionately representative sample of failure modes, BIT design features, and ATE design features are selected to be exercised. The number of tasks to be demonstrated should be determined using MIL-STD-105 methods based upon the size of the failure population and the desired test level.

2.2 Testability Demonstration Procedures.(1)

The plan to conduct the testability demonstration can be developed in accordance with the following paragraphs:

a. Item-related information. The plan should contain a list of all items to be demonstrated, and should contain for each item:

   (1) The identification of quantitative testability requirements to be demonstrated.

   (2) A list of documentation required prior to the start of the demonstration.

   (3) The mechanism for inserting failures into the item.

   (4) The procedure for applying stimulus.

   (5) The procedure for measuring and observing the test results.

b. General information. The plan should contain the following general information:

   (1) The identification of the demonstration site, contractor organization and responsibilities, and government responsibilities.

   (2) The membership and duties of the test team.

   (3) The methodology used in defining the failure population for the task pool.

   (4) The size of the task pool.

   (5) The procedure for selecting failures from the task pool for insertion into the items.

   (6) A listing of all support items required to calibrate equipment, insert failures, apply tests and observe responses.

   (7) A methodology for the interpretation and analysis of results (i.e., fault detected with isolation, fault detected without isolation, fault not detected, etc.)

   (8) The establishment of acceptance criteria for the demonstration (fault coverage, degree of fault isolation, MIL-STD-471 test method to be used).

(9) A plan for taking corrective action as may be needed.

(10) The schedule for demonstration, including identification of any other testing being performed concurrently and its interface with this test.

c. False Alarm Rate Demonstration Information.

(1) False alarm test information. For the conduct of the false alarm rate demonstration as required by the contract, the plan should identify the tests to be used as data sources, and the MIL-STD-781 Test Plan to be used.

(2) False alarm rate demonstration. Plan to demonstrate the achievement of the specified false alarm rate as required by the quality provisions of the contract. The false alarm rate, expressed in terms of average number of false alarms/hour of equipment or system operating time, should be demonstrated from false alarm data resulting from controlled tests (e.g., reliability demonstration tests, performance tests). The contractor and the procuring activity should jointly determine the specific data sources to be utilized, and the contractor should prepare a sub-plan as part of the Testability Demonstration Plan defining the procedures to be utilized in the collection and documentation of such data. Allowance may be made to continue to collect and record false alarm data through operational testing if necessary to identify design deficiencies.

(3) Demonstration criteria. The False Alarm Rate Demonstration should be based upon the criteria of MIL-STD-781, Reliability Tests. The MIL-STD-781 test plan to be used should be based upon the time available for test, acceptable decision risks, and specified discrimination ratio. The cumulative period of operating time must, as a minimum, include the operating time duration of the reliability demonstration test(s). Each confirmed false alarm should be treated as a relevant failure. The specified false alarm rate should be input to the test plan as a component of the total failure rate. The equipment should be considered acceptable with respect to false alarm rate if the acceptance criteria of the selected MIL-STD-781 test plan is met.

(4) Related false alarms. If two or more observed false alarms are positively attributable to a single design problem which has been identified for correction, only one false alarm is chargeable to the false alarm count.

d. Corrective Action. Planning for failure to pass the testability demonstration should require identification of appropriate redesign efforts by the contractor. Provide time in the development schedule to allow for correction of deficiencies and repeating of failed tests. Redesign considerations should include:

(1) Redesign of prime equipment.

(2) Redesign of test equipment.

(3) Redesign of interface devices.

(4) Redesign of built-in-test circuits.

(5) Redesign of diagnostic software.

(6) Redesign of ATE software.

(7) Correction of maintenance documentation.

(8) Correction of models used for testability analysis.

e. Other test data. In addition to formal demonstration, plan to maintain a record of all failures of assembled equipment during all testing throughout the contract. At each occurrence of failure the built-in-test function should be exercised, if applicable, and an entry on the failure report made to indicate compliance or non-compliance with the requirements for failure detection and isolation. A description of the malfunction, including failure symptoms, should be provided and documented in the Testability Analysis Report.

2.3 Testability Demonstration Plan Data Item Description.(14)

A Data Item Description (DID) has been written and proposed as the standard for documenting the testability demonstration plan. The following excerpt from that DID supplements the foregoing discussion of the contents of the plan.

a. *Description.* This plan describes the contractor's procedures for conducting the testability demonstration and is used by the procuring activity to evaluate the procedures. The Testability Demonstration may be used to evaluate the effectiveness of built-in hardware, diagnostic software, and/or design for testability features.

b. Application. This Data Item Description is applicable to development contracts during the full scale development phase which require the formal demonstration of testability requirements in systems and/or Configuration Items. This plan is not required if the testability demonstration is an integral part of the overall Qualification Testing and is adequately documented in the Qualification Test Plan.

c. Preparation Instructions. The Testability Demonstration Plan should contain the plans and procedures for conducting a demonstration of the effectiveness of built-in-test and design for testability features, and the validity of the Testability Analysis Model as required in MIL-STD-XXX.

The Plan should contain a list of all items to be tested, and should contain as a minimum for each item:

(1) A listing of qualitative and quantitative testability requirements.

(2) The identification of those qualitative and quantitative testability requirements to be demonstrated.

(3) The methodology for demonstrating the validity of the models used for testability analysis.

(4) A list of documentation required prior to the start of the demonstration (stimulus/response data, response/fault correlation data, testability analysis models, diagnostic software design documentation, etc.).

(5) The identification of the demonstration site, contractor organization and responsibilities, and government responsibilities.

(6) The methodology used in defining the failure population to be used during demonstration (considering function, component count, component failure rate, criticality of failure, etc.). The failure population chosen should maximize the information gained on the testability of the item.

(7) The mechanism for inserting failures into the item (e.g., prefaulted modules, jumper wires, simulated failures at module pins, etc.).

(8) The procedure for selecting failures from the demonstration failure population for insertion into the item.

(9) The procedure for applying stimulus (e.g., operational inputs, diagnostic software execution, tester stimulus, etc.) to the faulted item.

(10) The procedure for measuring and observing the test results (error branching, tester measurements, etc.).

(11) A listing of all support items required to insert failures, apply tests, and observe responses.

(12) A methodology for the interpretation and analysis of results (i.e., failure detected with isolation, failure detected without isolation, failure not detected, etc.).

(13) The establishment of acceptance criteria for the demonstration (% fault coverage, degree of fault isolation, time to test, false alarms) in accordance with MIL-STD-XXX

(14) A plan for taking corrective action as may be needed.

(15) The documentation of demonstration results in the Testability Analysis Report.

3. Completion Criteria.

This task is completed when the government has agreed to the details as set forth in the plan and has accepted the Testability Demonstration Plan in accordance with the terms of the CDRL.

TESTABILITY TASK COMPENDIUM

Task Reference Number F5B

PHASE: FULL SCALE DEVELOPMENT

FUNCTION: Testability Demonstration

TASK TITLE: Conduct Testability Demonstration

TASK OBJECTIVE: Demonstrate achievement of the testability design goals.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● Design Engineering	● HW BIT Data, $\overline{T}$ Design Data, HW Availability Schedule	● Demonstration Test HW Resource Requirements
● Application Software	● SW BIT and Diagnostic Test Data, SW Availability Schedule	● Demonstration Test SW Resource Requirements
● Support Equipment	● Test Equipment Availability Schedule	● Test Equipment Requirements
● Test Engineering	● Test Resource Data	● Demonstration Test Resource Requirements
● Program Management	● Schedule	● Equipment and Personnel Requirements

COST TRADEOFF INTER-RELATIONSHIPS: Not relevant to this task.

TASK SYNOPSIS:

1. Task Requirements.

1.1 Testability Demonstration.

The task requirement is to conduct a Testability Demonstration as required by the provisions of the contract and to demonstrate the achievement of specified (or goal) testability parameters on major end items.

2. Implementation.

Implementation should be carried out in accordance with the testability demonstration plan prepared as described in Full Scale Development Task 15. The demonstration plan should contain the method for conducting the demonstration test, the list of skills required to run the tests, the tasks pool, the corrective action plans, and the support/test equipment necessary for the test.

Results should include records of the test data on the testability demonstration data sheets (see Table 1).(1) Calculate the testability parameters from the data recorded on the data sheets. Compare the calculated parameters against the specified parameters to test for acceptability test data.

Combining the Testability demonstration with the Maintainability demonstration may prove cost- and schedule-effective. It is also prudent to dry-run the procedures prior to formal demonstration.

2.1 Demonstration Process.

The Testability Demonstration should be conducted in an environment which simulates, as closely as practicable, the test environment planned for the item. This environment should be representative of the support equipment, facilities, and technical data that would be required at the maintenance levels defined.

2.2 Built-in Test.

Each task is demonstrated by inserting the failure condition, running operational sequences and/or self-test sequences, and recording the results:

a. Each task should be analyzed to determine whether or not a clear indication of equipment failure is provided.

b. Each task should be analyzed to determine the level of ambiguity to which the equipment built-in test, external test equipment or manual test procedures perform the initial isolation. The results should be entered on the Testability Demonstration Data Sheet.

2.3 External Test.(1)

Modules identified by BIT as having possible failures should be removed from the prime equipment and exercised on external tester(s):

a. Each module test should be analyzed to determine whether or not a correct PASS/FAIL indication is provided.

b. Each correctly FAILING test should be analyzed to determine the level of ambiguity to which the test performs secondary isolation. The results should be entered on the Testability Demonstration Data Sheet.

TABLE 1. SAMPLE TESTABILITY DEMONSTRATION DATA SHEET

DEMO. NO.	TASK POOL NO.	FAILURE EVENTS						BIT TIME	ATE TIME	COMMENTS
		DETECTION BY:			ISOLATION					
		SYSTEM	BIT	ATE	BY: SYSTEM TO: CI	BIT MODULE	ATE COMPONENT			
1	57	N	Y	Y	-	3	1	70s	25m	Includes PASS time for 2 mods
2	113	Y	Y	N	1	2	-	50s	20m	Cannot duplicate
3	2	N	N	-	-	-	-	-	-	Undetected
4	30	Y	N	Y	1	-	N	-	-	ATE isolated to wrong group
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)

2.4 Testability Demonstration Data Sheet.(1)

Entries are made on the Testability Demonstration Data Sheet (Table 1) as follows:

Column 1 is assigned a sequential demonstration task number.
Column 2 indicates the demonstration task pool number selected at random for this task. This number is the link to the detailed information on the simulated failure.
Column 3 indicates that the failure did (Y) or did not (N) cause observable and documented anomalies in the operational behavior of the system.
Column 4 indicates that the failure was (Y) or was not (N) detected by built-in test features.
Column 5 indicates that the module containing the failure failed as a UUT on the ATE and all other failure-free modules passed (Y); otherwise, (N). The ATE testing may be assisted by built-in test features within the module.
Column 6 indicates the size of the ambiguity group (in equipments/CIs) resulting from system test localization. If the system did not exhibit anomalous behavior (column 3 = N), column 6 is assigned a dash for "not applicable." If the system exhibited anomalous behavior but did not provide any localization information, column 6 is assigned an N.
Column 7 indicates the size of the initial ambiguity group (in modules) resulting from successful isolation by equipment-level/module-level BIT. If BIT did not successfully isolate the failure, column 7 is assigned an N. If BIT did not detect the failure (column 4 = N), column 7 is assigned dash for "not applicable." (Alternatively, column 7 indicates the number of modules which would be redplaced in reaching the failed module if the modules are replaced n-at-a-time in the exact order given in the organizational-level fault dictionary.)
Column 8 indicates the size of the secondary ambiguity group (in components) resulting from successful isolation by the ATE system. The ATE testing may be assisted by module-level BIT. If ATE did not successfully isolate the failure, column 8 is assigned an N. If ATE did not detect the failure (column 5 = N), column 8 is assigned a dash for "not applicable." (Alternatively, column 8 indicates the number of components which would be replaced in reaching the failed component if the components are replaced n-at-a-time in the exact order given in the fault dictionary.)
Column 9 is the time required for failure isolation by BIT, measured from the initiation of BIT to the correct determination of the module ambiguity group.
Column 10 is the time required for failure isolation by ATE, measured for each module from the initiation of testing to the correct determination of the component ambiguity group.
Column 11 is reserved for comments, including references to deficiencies and corrective actions.

2.5 Testability Parameter Calculations.(1)

The following testability parameters can be calculated from the data in the Testability Demonstration Data Sheet:

P_{DS} = proportion of sample failures which are detected by the operational system

$$= \frac{N_{DS}}{T} = \frac{\text{number of Y's in column 3}}{\text{total failures inserted}}$$

P_{DB} = proportion of sample failures which are detected by built-in test

$$= \frac{N_{DB}}{T} = \frac{\text{number of Y's in column 4}}{T}$$

P_D = proportion of sample failures detected

$$= \frac{N_D}{T} = \frac{\text{number of Y's in column 3 or 4}}{T}$$

P_{IS} = proportion of sample system detections which reduce equipment/CI ambiguity size

$$= \frac{\text{number of "numbers" (non-dash,non-N) in column 6}}{N_{DS}}$$

P_{IB} = proportion of sample BIT detections which reduce module ambiguity size

$$= \frac{\text{number of "numbers" in column 7}}{N_{DB}}$$

P_{DA} = proportion of sample failures which are detected by ATE

$$= \frac{N_{DA}}{N_D} = \frac{\text{number of Y's in column 5}}{N_D}$$

P_{IA} = proportion of sample ATE/module-level BIT detections which reduce component ambiguity size

$$= \frac{\text{number of "numbers" in column 8}}{N_{DA}}$$

$I_S(K)$ = sampled degree of localization to K equipments provided by the operational system

$$= \frac{\text{count of times that column 6} = K}{N_{DS}}$$

$I'_S(L)$ = sampled degree of localization to L or fewer equipments provided by the operational system

$$= \sum_{K=1}^{L} I_S(K)$$

$I_B(K)$ = sampled degree of isolation to K modules provided by built-in test

$$= \frac{\text{count of times that column 7} = K}{N_{DB}}$$

$I'_B(L)$ = sampled degree of isolation to L or fewer modules provided by built-in test

$$= \sum_{K=1}^{L} I_B(K)$$

$I_A(K)$ = sampled degree of isolation to K components provided by ATE (and module BIT)

$$= \frac{\text{count of times that column 8} = K}{N_{DA}}$$

$I_A'(L)=$ sampled degree of isolation to L or fewer components provided by ATE (and module BIT)

$$= \sum_{K=1}^{L} I_A(K)$$

$I'_S(L)$, $I'_B(L)$, and $I_A'(L)$ are calculated for each value of L for which there is a fault isolation requirement specified.

2.6 Accept/Reject Criteria.(1)

a. Proportions. The observed proportions previously calculated should be compared against the corresponding specified parameter or against the corresponding parameters predicted by the testability models, according to contract requirements. The specified or predicted value should be considered to be achieved if the observed parameter, P, and the specified, P_s, have the following relationship

$$P \geq P_s - Z_C \sqrt{\frac{P(1-P)}{T}}$$

where T is the total number of relevant tasks in the sample and Z_C is the confidence level coefficient given below.

Z_C	PRODUCER RISK
0.84	20%
1.28	10%
1.65	5%

b. Failure isolation times. The observed failure isolation time for built-in test for each demonstration task is taken from column 9 of Table I. For those tasks which did not result in successful isolation of the failure, the failure isolation time using manual troubleshooting methods should be estimated and used as the observed time. Appropriate methods of MIL-STD-471 are used, substituting observed isolation times for observed maintenance times, to determine if the specified mean (or maximum) failure isolation time is met:

Mean specified:	Use method 1
Percentile specified:	Use method 2
Both specified:	Use method 8

2.7 False Alarm Rate Demonstration Criteria.(1)

The False Alarm Rate Demonstration should be based upon the criteria of MIL-STD-781, Reliability Tests. The MIL-STD-781 test plan to be used should be based upon the time available for test, acceptable decision risks, and specified discrimination ratio. The cumulative period of operating time (T) must as a minimum, include the operating time duration of the reliability demonstration test(s). Each confirmed false alarm should be treated as a relevant failure. The specified false alarm rate should be input to the test plan as the Predicted MTBF. The equipment should be considered acceptable with respect to false alarm rate if the acceptance criteria of the selected MIL-STD-781 test plan is met.

a. Related false alarms. If two or more observed false alarms are positively attributable to a single design problem which has been identified for correction, only one false alarm is chargeable to the false alarm count.

2.8 Documentation.

The results of the demonstration should be documented for submittal to the government.

2.9 Testability Demonstration Pass/Fail.

Successful completion of the demonstration is indicated if a comparison made between the parameters calculated from the test data and the contractually required parameters shows that the accept criteria have been met.

3. Completion Criteria.

The task is complete upon government acceptance and approval of the demonstration report.

TESTABILITY TASK COMPENDIUM
Task Reference Number F6A

PHASE: FULL SCALE DEVELOPMENT

FUNCTION: Final Testability Analysis Report

TASK TITLE: Prepare Final Testability Analysis Report

TASK OBJECTIVE: Provide a single-document repository for all significant testability data.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● Design Engineering	● HW $\overline{T}$ Design Data, BIT data, HW Tradeoffs	● Testability Analysis Report
● Application Software	● SW $\overline{T}$ Features, SW Tradeoffs	
● Support Equipment	● ATE Interface Data	
● Test Engineering	● Test Program $\overline{T}$ Data	
● Life Cycle Costing	● Tradeoff Costing Data	

COST TRADEOFF INTER-RELATIONSHIPS: Not applicable to this task.

TASK SYNOPSIS:

1. Task Requirements

1.1 Report Data.(1)

Prepare a Final Testability Analysis Report that documents the testability program instituted and used. Specification data, tradeoffs/analyses results, demonstration data, and standing recommendations should be included within the body of the report, with cross-reference to source and substantiating data and documents. The testability analysis report should include the following information. (Previously submitted material which is unchanged may be referenced or its location identified.)

a. Testability Specifications. System/CI testability specifications approved at PDR and CDR, plus any changes approved during Full Scale Development.

b. Testability Tradeoffs. Documentation of the impact of operational requirements on testing, alternative designs, tradeoff analysis, and rationale for selection of design alternatives. Includes data presented at PDR, and CDR, plus any new analysis due to changes during Full Scale Development.

c. Qualitative Testability Analysis. The material submitted in the preliminary Testability Analysis Reports at PDR and CDR, updated to reflect any functional design changes during detail design.

d. Quantitative Testability Analysis, ATE Testing. The UUT testability analysis data, for each type of Unit Under Test, including:

(1) Definition of the failure population in accordance with the specified failure modes.

(2) Identification of the test stimulus, including built-in stimulus and external stimulus.

(3) Determination of the percentage of failures in the failure population which are detected by the test stimulus.

(4) Determination of the level of fault isolation achievable.

(5) Justification for classes of faults remaining undetected or which are poorly isolated.

e. Quantitative Testability Analysis, Built-In Testing. The testability analysis data for each configuration item including:

(1) Definition of the failure population in accordance with the specified failure modes.

(2) Identification of the built-in-test stimulus.

(3) Determination of the percentage of failures in the failure population which are detected by the test stimulus.

(4) Determination of the level of fault isolation achievable.

(5) Justification for classes of faults remaining undetected or which are poorly isolated.

f. Quantitative Testability Analysis, System Level. The overall testability analysis data, including:

(1) A description of the integration of testing for each item at the system level.

(2) A determination of the overall system failure coverage and level of fault isolation.

    (3) An estimate of developmental and recurring costs associated with design for testability.
    (4) An estimate of developmental, production, and support savings associated with design for testability.
    (5) An estimate of system readiness improvement attributable to design for testability.

g. Testability Demonstration Data. Summary data for each item involved in testability demonstration including original plans, summarized results and any corrective action taken. The detailed demonstration procedures and raw test results need not be included.

h. Recommendations. Recommended action to be taken to remedy testability deficiencies or improve the level of testability achievable through prime equipment engineering changes, ATE improvements and/or Test Program Set improvements.

2. Implementation.

The preliminary testability analysis completed in the Validation Phase (Task Reference Number V5A) contains much of the prelininary data needed for the final report. The final testability analysis report is arrived at by updating the initial data, then incorporating essential information drawn from those testability tasks completed in the Full Scale Development Phase.

2.1 Report Preparation Instructions.

A Data Item Description (DID) has been written and proposed as the standard for preparation of the Testability Analysis Report. The following is a paraphrased excerpt from that DID.(9)

a. Recommended for inclusion in the qualitative sections of the Testability Report include:

    (1) A description of the partitioning used to enhance testability in accordance with MIL-STD-XXX.
    (2) A description of each applicable item (i.e., potential or actual UUT).
    (3) An analysis of potential failure modes and effects for each item. Data on failure rates and confidence levels may be referenced from the Reliability Program, as is applicable.
    (4) A summary of the overall maintenance concept taken from Maintainability Program and ILS Plan. A description of the overall test strategy to implement the maintenance concept, including coordination between BIT and ATE.

(5) A description of the test strategy to be used for each applicable item, as determined by the overall test strategy.

(6) A functional description of built-in-test features, including hardware and software BIT, and testability features, including controllability and observability considerations, for each item.

(7) A functional description of testability measurement techniques to be used, including computer-aided analysis tools.

b. Recommendations for inclusion in the quantitative sections of the Testability Analysis Report include:

(1) For each item defined in 2.1a(2) above, a description of the Testability Analysis Model including:

- A definition of the failure population in accordance with the specification.
- Identification of the test stimulus, including built-in-test stimulus and external stimulus.
- A determination of the percentage of failures in the failure population which are detected by the test stimulus.
- A determination of the level of fault isolation achievable with the test stimulus.
- The justification for classes of failures remaining undetected or which are poorly isolated.

(2) For the overall system:

- A description of the integration of the items and their test stimulus/response at the system level.
- A determination of the overall system fault coverage and level of fault isolation based upon an appropriate combination of these characteristics for each item.
- An estimate of developmental and recurring costs associated with design for testability, including weight, volume, and reliability penalties, and increased computer memory requirements.
- An estimate of developmental, production, and support savings associated with design for testability, including reduced checkout time, training, spares, and ATE costs.

c. The testability analysis report shall include the following support data. This support data shall be included within the body of the report.

The Testability Analysis Support Data shall reflect each item's current configuration and include:

(1) Testability Analysis Model for the item:

- Component characteristics
- Component failure models
- Component interconnection data
- Test control nodes
- Test observation nodes
- Interrelationship of models

(2) Testing Data for the item:

- Test stimulus
- Predicted good response
- Predicted fault responses
- Test stimulus restrictions
- Test response tolerances
- Fault coverage data
- Fault isolation data

2.2 Completeness.

The Testability Analysis Report may be contractually required by CDRL/DID. The degree of acceptability can be enhanced by:

- Adhering to the DID requirements
- Using the topic checklist under 2.1, Report Data, as a gauge of completeness
- Review by writing services quality assurance functions of the completed report for adherence to writing principles, clarity and conciseness.

3. Completion Criteria.

Successful completion of this task occurs upon government acceptance and approval of the Testability Analysis Report.

TESTABILITY TASK COMPENDIUM
Task Reference Number F7A

PHASE: Full Scale Development

FUNCTION: Category/Developmental/Operational Testability Evaluation*

TASK TITLE: Monitor/Evaluate/Propose corrective action

TASK OBJECTIVE: Exploit CAT/DT/OT to obtain early assessment of testability effectiveness.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIP:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● Design Engineering	● HW Design Test Requirements	● HW Related Corrective T Action
● Application Software	● SW Design Test Requirements	● SW Related Corrective T Action
● Test Engineering	● Test Data	
● Reliability Engineering	● Data and Consultation	● Data and Consultation
● Maintainability Engineering	● Data and Consultation	● Data and Consultation
● Life Cycle Costing	● Proposed Corrective Action Cost Tradeoffs	● Testability Data Supporting LCC

COST TRADEOFF INTER-RELATIONSHIPS: The Life Cycle Costs associated with corrective actions should be evaluated. The analysis required is largely one of costs and penalties of implementing change, compared to cost savings and benefits realized from the change.

* The major source document for this task used the term "Operational Test and Evaluation" which is the term also used in the task synopsis. The term may be construed to mean also Category testing or Developmental Testing.

TASK SYNOPSIS:

1. Task Requirements.

The task consists essentially of data access and evaluation.

a. Data collection. Establish a system to access or collect the data required for the evaluation of testability phenomena occurring during the Operational Test and Evaluation of the system or equipment. Data for analysis of confirmed failures and false alarms is of high significance.

b. Data analysis. Develop a methodology for analysis of the operational effectiveness of the testability design.

c. Corrective action. Where the analysis projects future likelihood of non-attainment of testability goals or other problems relating to testability, develop means of proposing and evaluating modifications to the prime equipment, test equipment, software or other test program element as appropriate.

2. Implementation.(1)

The objective of the testability evaluation is to evaluate the impact of actual operational and maintenance environments on the ability of production equipment to be tested. The effectiveness of testability design techniques for intermediate or depot level maintenance tasks should be monitored and evaluated as part of the operational testability evaluation. The maintenance tasks to be evaluated should be limited to those resulting from actual operational problems and maintenance actions.

2.1 Data Collection.(1)

A procedure to collect or otherwise access data must be established prior to the operational test, preferably as an adjunct to that used for data collection by the maintainability group or by the reliability group. The data collected should include a description of all operational anomalies and maintenance actions. In addition, the data system should be compatible with existing data systems in use by the user organization.

a. Tracking of confirmed failures. Data for each confirmed failure should be compiled to address the following questions:

(1) Built-in Test.

(a) Did built-in test detect the failure?

(b) Did built-in test correctly indicate which mission functions were lost?

(c) Did built-in test provide accurate fault isolation information for corrective maintenance actions?

(d) What was the ambiguity size (number of modules to be removed or further tested) due to fault localization/ isolation by built-in test?

(e) How much time was required for fault isolation at the organizational level of maintenance?

(2) ATE Testing.

(a) Were any workarounds required to overcome mechanical or electrical deficiencies in the UUT/ATE interface?

(b) Was the documentation for UUT hookup and power-up procedures accurate?

(c) Did the ATE system provide failure detection results consistent with those of the initial failure detection by BIT?

(d) Did the ATE system (in conjunction with any module BIT) provide accurate fault isolation?

(e) Were the test results repeatable?

(f) Was the observed test result documented in the maintenance documentation?

(g) Was the failed component listed under the observed test result in the maintenance documentation?

(h) What was the ambiguity size (number of components to be removed or further tested) due to fault isolation by the ATE system?

(i) How much time was required for fault isolation?

(j) How much total time (calendar time) was required to repair the module?

b. Tracking of false alarms. Data collected should include a description of all cases in which the built-in test reported the detection of a failure but with no operational anomalies present and with no faults subsequently identified. The data should also document cases in which an ATE test program declared a failure in a UUT with no faults subsequently identified within the UUT. The data should be grouped by alarm type and address the following questions:

(1) What is the characterization of the alarm type?

(2) What is the frequency of occurrence of the alarm?

(3) What failure or failures are expected to cause the observed alarm?

(4) What are the potential consequences of ignoring the alarm (in terms of crew safety, launching unreliable weapons, etc.)?

(5) What are the operational costs of responding to the false alarm (in terms of aborted missions, degraded mode operation, system downtime)?

(6) What are the support costs associated with a false alarm?

(7) What additional data is available from operational software dumps? (e.g., soft failure occurrences, branch histories, interrupt histories, etc.)?

(8) Has the system environment (or the understanding of the system environment) changed since the system's tolerances or transient characteristics were specified?

2.2 Analysis.

The literature provides little specialized guidance for analysis of results, and the implementation is that of generalized problem solving. The basic purpose of the analysis is to "develop a tailored methodology for the analysis of operational problems so as to ascertain if built-in test hardware and software, ATE hardware and software, and maintenance documentation are meeting specifications in terms of fault coverage, fault resolution, false alarms, fault detection times and fault isolation times. The testability models* and analysis should be used as the basis for evaluating the effectiveness of any proposed modifications to prime equipment or test equipment." (1) The following hints for conduct of analysis are drawn largely by inference from related source material.

a. Work objectively with data elements accessed or collected as noted above.

b. Separate testability issues from those which are more properly of a reliability, maintainability or other nature; conduct liaison with the other disciplines to assure appropriate treatment of all data.

---

*See Task V4D (Ed.)

c. Ascertain conditions unique to the test environment which may influence the analysis as projected to the true future operating environment.

d. Orient the analysis to determination of the extent to which the system meets the T requirements and/or goals and to correction of true T potential problems at the earliest possible time to provide the least complex and most cost-effective solutions.

e. Merge data and analyses with the parallel results from contractor and government development testing and from testability, reliability and maintainability demonstrations.

f. Organize and document the presentation of analysis results to be objective and to lucidly support the recommendations.

g. Use the guidance contained in the Testability Task Compendia of functions V4, V5 and V6 to aid in structuring the analysis.

h. Refer to Appendix A for excerpts from the literature which may aid in false alarm analysis and BIT effectiveness evaluation.

2.3 Corrective Action.

As is the case with analysis, corrective action implementation relies on general problem solving techniques. The following hints are drawn by inference from related source material.

a. The analysis should evolve completely to the identification of optimum solutions. The urge to correct problems by re-design should be resisted until it is proven that no other alternatives exist.

b. Coordinate each recommendation with all other disciplines affected by the specifics of the recommendation.

c. Include with each recommendation a cost analysis treating costs and benefits for the entire life cycle.

3. Completion Criteria.

This task may have indefinite closure where its recommendations, in whole or in part, require long term action plans. The task may be considered as completed after government approval of the analysis report and recommendations and after assumption of responsibility, by a management entity, for any remaining testability-related action items.

APPENDIX F7A1

1. False Alarm Causes.(15)

False alarms occur for several reasons: a) the system under test does not operate as expected, b) the test mechanization is incorrect, c) the software is programmed incorrectly, d) failure limits have been set too tight for dynamic conditions, e) test system logic circuitry fails, or f) test parameters are affected by noise. .....these false indications ..... (require) detailed investigation of the system performance and coordination with the design personnel, system supplier, and .....test personnel to effect an optimum resolution.

2. A BIT Evaluation Technique.(16)

BIT evaluation may be aided by use of the following technique, which is based on a variation of a simple truth table.

TABLE 1

BIT INDICATION MATRIX

BIT INDICATION	ACTUAL FAILURE	NO FAILURE
→	(correct)	(false)
NO - GO	a*	c*
→	(false)	(correct)
GO	b*	d*

*The letters are notations for terms used in the following equations:

(1) Fault Detection (FD): The MIL-STD-1309B definition of fault detection is - "One or more tests performed to determine if any malfunctions or faults are present in a unit." In equation form and relating back to Table 1, detection as a measure of BIT effectiveness can be expressed as:

$$\text{BIT/FT (\%)} = \frac{a}{a + b} \times 100\%$$

(2) False Alarm Rate (FAR): The MIL-STD-1309B definition of false alarm is - "An indicated fault where no fault exists." In equation form, FAR can be expressed as:

$$\text{BIT/FAR (\%)} = \frac{c}{a+c} \times 100\%$$

(3) Fault Isolation (FI): The MIL-STD-1309B definition of fault isolation is - "Tests performed to isolate faults within the unit under test," and is an equation for a given sample it is:

$$\text{BIT/FI (\%)} = \frac{\text{Faults correctly isolated by BIT}}{a} \times 100\%$$

TESTABILITY TASK COMPENDIUM

Task Reference Number F8A

PHASE: FULL SCALE DEVELOPMENT

FUNCTION: Final Preproduction Readiness Review

TASK TITLE: Provide field testability data

TASK OBJECTIVE: Assure that optimum testability is contained in production designs and that timely actions are continuing to resolve outstanding corrective actions.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

Coordination is appropriate with all other disciplines participating in the program. The inter-relationships are those of all the functions and tasks of the FSD phase.

COST TRADEOFF INTER-RELATIONSHIPS:

Underlying inter-relationships are those of all the functions and tasks of the FSD phase which are relevant to any action items still in resolution or completion stages at the time of the final review.

TASK SYNOPSIS:

1. Task Requirements.

1.1 This review may be formal or informal, serving if needed to support CDR-type activity (where the system readied for production is in turn a subsystem of a larger or more comprehensive system) or simply for internal recapitulation of the testability design for benefit of the testability participants only. It might also support DSARC (or lower level SARC) activity.

2. Implementation.

The concept for implementation is the same as that for Task 14 (CDR Support), i.e., preparation, presentation and followup.

2.1 Preparation.

Preparation for the review consists of review of all testability materials and data, review of the testability data collected during the operational test and evaluation, preparation of recommendations to improve testing, and organization of the major testability elements for concise, unambiguous presentation.

2.2 Presentation.

The testability engineer may present summary information from the testability analysis report, from the testability demonstration, and from the operational test and evaluation. All necessary support data must also be prepared for possible review.

2.3 Participation.

The testability engineer must also participate in the review, revision, and approval process of the overall readiness review.

3. Completion Criteria.

3.1 Government approval of the Testability Analysis Report and the system/equipment specifications, together with approval of review minutes comprise the measure of completion. Action item responsibility may be transitioned to another form of continuing management attention.

FULL SCALE DEVELOPMENT PHASE BIBLIOGRAPHICAL REFERENCES

1. A Study of Testability Standardization for Electronic Systems and Equipment prepared by Testing Technology Office, NSWC & Command & Control Applications Branch, NOSC for NESC, Code 304, pg. 20, Appendix B, 37.
2. The Test Requirements Document A Key to Improved Electronic Support by E.E.Johnson, Jr.; Industry/Joint Services Automatic Test Conference & Workshop Proceedings, San Diego, Apr 78, pg. 363.
3. MIL-STD-1519(USAF), Test Requirements Document, Preparation of. NOTICE 1, 1 August 1977.
4. DI-E-30139 Computer Program Development Specification.
5. Test Programming Languages by Damon C. Hart; Electronics Test, June 1980, pg. 48.
6. MIL-STD-2077(AS) Test Program Sets, General Requirements for. 9 March 1978
7. Cost Estimating Relationships for Automatic Test Equipment Test Program Set Development by P.M. Toscano; Industry/Joint Services Automatic Test Conference & Workshop Proceedings, San Diego, Apr 78, pg. 366.
8. Selection Guide for Digital Test Program Generation Systems, NAVMATINST 3960.9A dtd 19 September 1979.
9. Life Cycle Costs of Test Program Sets by D.J. Zingg; Industry/Joint Services Automatic Test Conference & Workshop Proceedings, San Diego Apr 78, pg. 458.
10. Small Digital Module Testers for Shipboard Use (DIMOTE) by G. Margulies, NAVELEX 4804; Final Report Usability and Cost Effective Selection of Test Equipment prepared for Naval Electronics Systems Engineering Center San Diego, CA, Contract No. N00123-75-C-0910.
11. Economics of Failure Detection in ATE by K.R. Hilberth, GT&T Industries, Inc., Van Nuys, CA; Industry/Joint Services Automatic Test Conference & Workshop Proceedings, San Diego, Apr 78, pg. 513.
12. Digital Automatic Test Program Generation Selection by D.B. Day, Support Systems Associates, Inc.; CH 1488-6/79/0000-0295 @ 1979 IEEE, pg. 295.
13. Availability/Operational Readiness Test Subsystem Cost Tradeoffs, RADC-TR-80-182, pg. 37.

14. A Framework for Designing Testability into Electronic Systems by W.L. Keiner; TR3826, Naval Surface Weapons Center, Dahlgren Laboratory (Code K43), Dahlgren, VA 22448, pg. 42, 53.

15. Onboard Test System Design Guide, Rockwell International, North American Division, Los Angeles, CA; TFD-80-206.

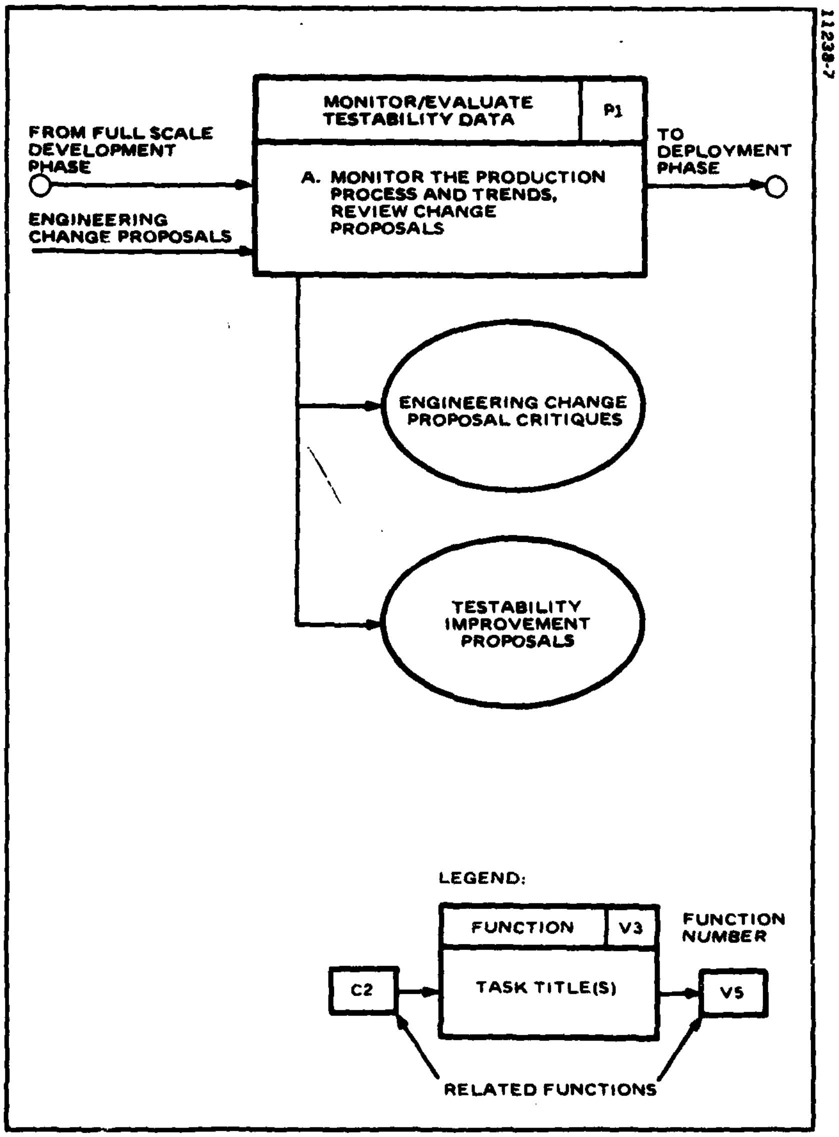
16. BIT/SIT Improvement Project (Phase 1): Evaluation of Selected USAF Aircraft BIT/SIT Systems/Subsystems, ASD-TR-79-5013, pg. 47, 49.

TESTABILITY TASK COMPENDIA

SECTION IV - PRODUCTION PHASE

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS



Production Phase Testability Tasks

TESTABILITY TASK COMPENDIUM
Task Reference Number P1A

PHASE: PRODUCTION

FUNCTION: Monitor/Evaluate Testability Data

TASK TITLE: Initiate and/or review change proposals

TASK OBJECTIVE: Assure that testability is taken into account in the change proposal cycle.

DESIGN DISCIPLINE TESTABILITY ($\overline{T}$) INTER-RELATIONSHIPS:

$\overline{T}$ INTERFACE	INPUT TO $\overline{T}$	OUTPUT FROM $\overline{T}$
● Maintainability Engineering	● Maintainability Evaluation Data	● Evaluation of Data, Suggestions for $\overline{T}$ Improvements, Initiation of Engineering Change Proposals as Required
● Producibility Engineering	● Manufacturing Data	
● Test Engineering, Factory & Customer	● Test Results Data	
● Training	● $\overline{T}$ Relevant Data	
● Change Review Board	● Non-$\overline{T}$ ECP for Evaluation	● $\overline{T}$ ECP for Evaluation

COST TRADEOFF INTER-RELATIONSHIPS: Production analysis may involve any of the cost tradeoffs encountered in the development phases. Actual or accrued cost data should be accessed whenever available. Parameters used for future cost estimating should be kept current in the light of observations made of actual production.

TASK SYNOPSIS:

1. **Task Requirements.**

The task is two-fold: first, to sustain the integrity and consistency of the system's testability characteristics in the process of implementing ECP's and, secondly, to enhance testability where feasible and cost effective via the ECP process. Access to or collection of factual data is necessary to perform analysis which objectively support all comments and recommendations.

2. Implementation.

Implementation consists of data collection, reduction and analysis followed by identification of corrective actions. The testability engineer's recommendations for courses of action need to be supported by objective conclusions drawn from the data analysis as do his comments on those courses endorsed by other disciplines.

2.1 Data Collection.[1]

a. From manufacturing processes, cognizance can be maintained of failures of assembled equipment during factory and/or acceptance testing throughout the production contract. At each failure occurrence, exercise of the built-in-test function as applicable can provide, compilations of the following data:

(1) degree of compliance with the requirements for failure detection and isolation.

(2) description of the malfunction including failure symptoms.

(3) built-in-test role:
- did BIT detect the failure?
- did BIT correctly indicate the functions lost?
- did BIT correctly isolate the fault?
- what was the ambiguity size due to BIT fault isolation?

(4) description of all cases in which BIT reported a failure but no faults were subsequently identified (false alarms).

b. From developmental tests and operational tests, a testability data collection system can be established that is integrated as much as possible with similar requirements (such as reliability and maintainability) and is compatible with existing customer data systems. Data can be compiled for each confirmed failure that addresses the following questions:

(1) Built-in-test role:
- did BIT detect the failure?
- did BIT correctly indicate which mission functions were lost?
- did BIT provide accurate fault isolation information for corrective maintenance actions?
- what was the ambiguity size (number of modules to be removed or further tested) due to fault localization/ isolation by BIT?
- how much time was required for fault isolation at the organizational level of maintenance?

(2) ETE (Including ATE) Testing

- were any workarounds required to overcome mechanical or electrical deficiencies in the UUT/ETE interface?
- was the documentation for UUT hookup and power-up procedures accurate?
- did the ETE system provide failure detection results consistent with those of the initial failure detection by BIT?
- did the ETE system (in conjunction with any module BIT) provide accurate fault isolation?
- were the test results repeatable?
- was the observed test result documented in the maintenance documentation?
- was the failed component listed under the observed test result in the maintenance documentation?
- what was the ambiguity size (number of components to be removed or further tested) due to fault isolation by the ETE system?
- how much time was required for fault isolation?
- how much total time (calendar time) was required to repair the module?

(3) False alarms

- what is the characterization of the alarm type?
- what is the frequency of occurrence of the alarm?
- what failure or failures are expected to cause the observed alarm?
- what are the potential consequences of ignoring the alarm (in terms of crew safety, launching unreliable weapons, etc)?
- what are the operational costs of responding to the false alarm (in terms of aborted missions, degraded mode operation, system downtime)?
- what are the support costs associated with a false alarm?
- what additional data is available from operational software dumps? (e.g., soft failure occurrences, branch histories, interrupt histories, etc.)
- has the system environment (or the understanding of the system environment) changed since the system's tolerances or transient characteristics were specified?

c. From demonstrations, the same data collection system developed for tracking factory failures (2.1.a) may be used during formal demonstrations (Testability, Reliability and Maintainability) and Qualification Testing.

d. From introduction into customer inventory, the following are existing maintenance data sources:

(1) Formal AFM 66-1 Maintenance Data Collection:(2) This system is the major AF-wide maintenance data collection system. It is a multi-purpose computerized data collection system that is designed to capture manhours expended in maintenance actions, when failures were discovered, how the equipment malfunctioned, type of repair action taken, type of failure, and dates of actions.

(a) Input Data.

- AFTO Form 350 (Reparable Item Processing Tag): This tag is filled out by the maintenance crew and attached to any LRU that is removed from an aircraft. Information included on this tag is the Work Unit Code (WUC), how malfunctioned code, when discovered code and a brief written description of the problem/malfunction. Information contained on this tag provides the direct input for the completion of the AFTO Form 349 in the intermediate shop.

- AFTO Form 349 (Maintenance Data Collection Record): This form is filled out at the intermediate shop for every LRU pulled from an aircraft and is assigned a Job Control Number (JCN). Some of the data that is included on this form by specific codes is the WUC of the equipment, when discovered, how malfunctioned, action taken and type maintenance performed. Also, space is provided to write a description of the discrepancy and corrective action taken. The specific codes entered on the form are the direct inputs to the computer reports.

(b) There are a number of possible computer outputs of the AFM 66-1 system. Some of the available outputs are:

- BLIS (Base Level Inquiry System): This output records maintenance activities of a particular base and is normally available only at the base. The BLIS reports are compiled from AFTO Form 349's and are available in various formats.

- AFLC D056 Reports: These are computerized reports processed by AFLC that are intended as measures of product performance and maintenance manhours and reflect areas that require improvement. AFLCR 66-15 outlines the different possible reports available.

- AFTO Form 95 (Significant Historical Data): This optional form can be filled out as a history of maintenance actions taken for a particular LRU every time the LRU comes into the intermediate shop for repair. Actual information displayed can vary from shop to shop. Normally, records are maintained by JCN, WUC and LRU serial number and include actual repair actions/ remedies, components replaced, adjustments made, Not Reparable This Station (NRTS) codes, Could Not Duplicate (CND) or bench check retest OK (RETOK) occurrences and test station downtime. When maintained accurately, these records provide a measure of the FAR and indicate which LRUs were removed for BIT fault detection or for other reasons, or both.

(2) USN Standard Maintenance and Material Management System (3M): This system is virtually parallel to the AFM 66-1 system, using USN form numbers and notations.

(3) The Army Maintenance Management System (TAMMS): The TAMMS also parallels the AFM 66-1 system.

2.2 Data Reduction and Analysis.

a. A methodology for data reduction and problem analysis relevant to testability should be developed so as to ascertain if built-in-test hardware and software, ATE hardware and software, and maintenance documentation are meeting specifications in terms of fault coverage, fault resolution, false alarms, fault detection time and fault isolation time.

b. The testability data reduction and analysis methodology may be extended to identify trends relevant to other hardware characteristics such as a change in component failure rate or increase in remove and replace times.

2.3 Corrective Action.

The data collection, reduction and analysis activity may identify testability related problems whose only solution is by ECP. The testability models and analyses developed during the validation phase and updated during full scale development may be used to justify the proposed modifications to the prime equipment or test equipment. For ECPs generated by groups other than testability, the testability models and analyses may be used to aid in evaluating the effectiveness of the proposed modification.

3. Completion Criteria.

The overall task of collecting data, analyzing data and taking corrective action will continue into the operation and support phase activity as demonstrations and testing are replaced by the build-up of operating inventory.

The task may be considered as complete when production has been completed and/or the government assumes engineering responsibility for the prime system. Alternatively, the responsibility may transition to the entity responsible for monitoring of operations and support (Function D1).

PRODUCTION PHASE BIBLIOGRAPHICAL REFERENCES

1. A Study of Testability Standardization for Electronic Systems and Equipment prepared by Testing Technology Office, NSWC & Command & Control Applications Branch, NOSC for NESC, Code 304, pg. 37.

2. BIT/SIT Improvement Project (Phase 1): Evaluation of Selected USAF Aircraft BIT/SIT Systems/Subsystems, ASD-TR-79-5013, pg. 47.

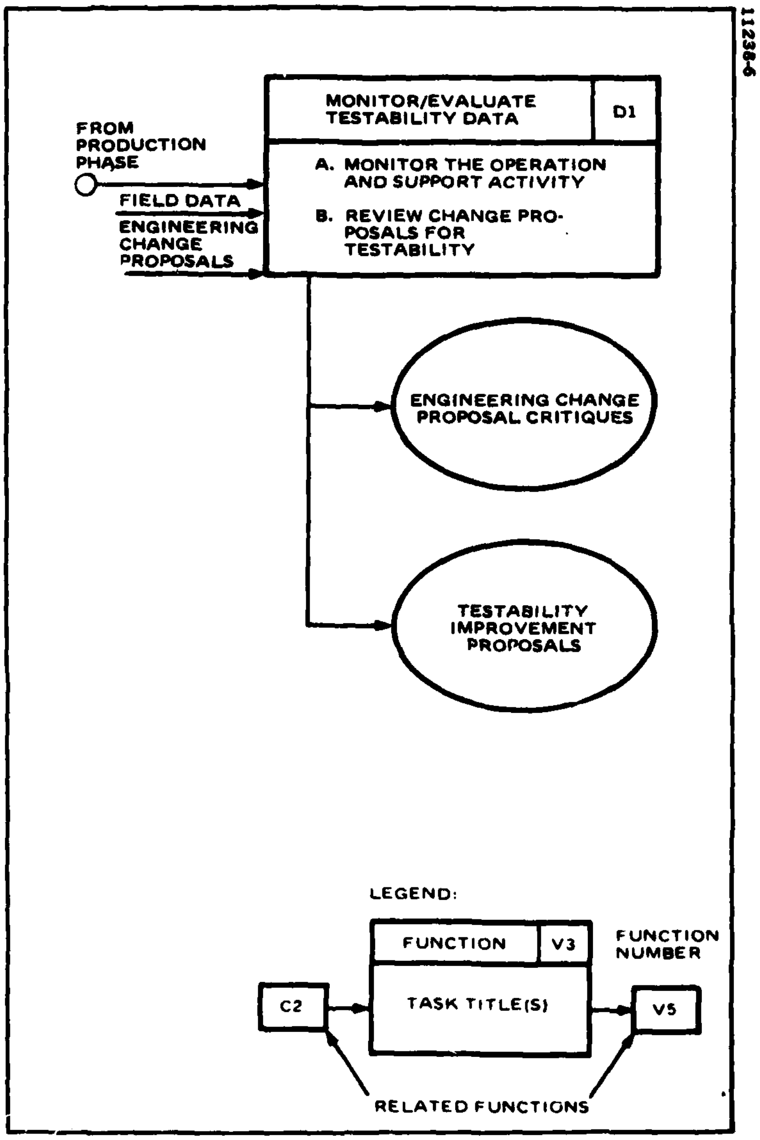
TESTABILITY TASK COMPENDIA

SECTION V - DEPLOYMENT PHASE (OPERATIONS AND SUPPORT)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES



Deployment Phase Testability Tasks

TESTABILITY TASK COMPENDIUM
Task Reference Number D1A

PHASE: OPERATION AND SUPPORT

FUNCTION: Monitor/Evaluate Testability Data

TASK TITLE: Monitor operation & support activity

TASK OBJECTIVE: Measure the achieved testability of the system for improving the current system and for application of knowledge to new and evolving designs.

DESIGN DISCIPLINE TESTABILITY ($\bar{T}$) INTER-RELATIONSHIPS:

$\bar{T}$ INTERFACE	INPUT TO $\bar{T}$	OUTPUT FROM $\bar{T}$
● Contractor Field Service	● Field Data	● Testability Trends, Engineering Change Proposals
● Government Data Systems	● Field Data	● Testability Trends, Engineering Change Proposals

COST TRADEOFF INTER-RELATIONSHIPS: Operations and Support Analysis may involve any of the cost tradeoffs encountered in development phases. Actual or accrued cost data should be accessed wherever available, and parameters used for future cost estimating should be kept current in the light of observations made of actual operations and support.

TASK SYNOPSIS:

1. **Task Requirements.**

To measure achieved Testability, it is necessary to collect or access in any way feasible all operational, maintenance, repair, test, and support data from all operations and maintenance levels to determine the effectiveness of testability design techniques for organizational, intermediate and depot level maintenance tasks.

1.1 Monitoring also provides the means to evaluate the impact of actual operational and maintenance environments on the testability of the production equipment.

2. Implementation.

2.1 A data collection system should be established as necessary to meet the needs of the testability evaluation. The data collected should include a description of all operational anomalies and maintenance actions. Data collection is to be integrated as much as possible with similar data collection requirements such as those for reliability and maintainability. In addition, the data system should be compatible with existing data systems in use by the military user.

2.2 Tracking of Confirmed Failures.⁽¹⁾

Data for each confirmed failure should be compiled to address the following questions:

a. Built-in Test.

- Did built-in test detect the failure?
- Did built-in test correctly indicate which mission functions were lost?
- Did built-in test provide accurate fault isolation information for corrective maintenance actions?
- What was the ambiguity size (number of modules to be removed or further tested) due to fault localization/isolation by built-in test?
- How much time was required for fault isolation at the organizational level of maintenance?

b. ETE/ATE Testing.

- Did the ETE/ATE system provide failure detection results consistent with those of the initial failure detection by BIT?
- Did the ETE/ATE system (in conjunction with any module BIT) provide accurate fault isolation?
- Were the test results repeatable?
- Was the observed test result documented in the maintenance documentation?
- Was the failed component listed under the observed test result in the maintenance documentation?
- What was the ambiguity size (number of components to be removed or further tested) due to fault isolation by the ETE/ATE system?

- How much time was required for fault isolation?
- How much total time (calendar time) was required to repair the module?

c. Tracking of false alarms. Data collected should include a description of all cases in which the built-in test reported the detection of a failure but with no operational anomalies present and with no faults subsequently identified. The data should also document cases in which an ATE test program declared a failure in a UUT with no faults subsequently identified within the UUT. The data should be grouped by alarm type and address the following questions:

- What is the characterization of the alarm type?
- What is the frequency of occurrence of the alarm?
- What failure or failures are expected to cause the observed alarm?
- What are the potential consequences of ignoring the alarm (in such terms as crew safety and launching unreliable weapons)?
- What are the operational costs of responding to the false alarm (in terms of aborted missions, degraded mode operation, or system downtime)?
- What are the support costs associated with a false alarm?
- What additional data is available from operational software dumps? (e.g., soft failure occurrences, branch histories, interrupt histories,)?
- Has the system environment (or the understanding of the system environment) changed since the system's tolerances or transient characteristics were specified?

2.3 Existing Maintenance Data Sources.

a. Formal AFM 66-1 Maintenance Data Collection.(2) This system is the major AF wide maintenance data collection system. It is a multipurpose computerized data collection system that is designed to capture manhours expended on maintenance actions, when failures were discovered, how equipment malfunctioned, type of repair action taken, type of failure, and dates of actions.

(1) AFTO Form 350 (Reparable Item Processing Tag): This tag is filled out by the maintenance crew and attached to any LRU that is removed from an aircraft. Information included on this tag is the Work Unit Code (WUC), how malfunctioned code, when discovered code and a brief written description of the problem/malfunction. Information contained on this tag provides the direct input for the completion of the AFTO Form 349 in the intermediate shop.

(2) AFTO Form 349 (Maintenance Data Collection Record): This form is filled out at the intermediate shop for every LRU pulled from an aircraft and is assigned a Job Control Number (JCN). Some of the data that is included on this form by specific codes is the WUC of the equipment, when discovered, how malfunctioned, action taken and type maintenance performed. Also, space is provided to write a description of the discrepancy and corrective action taken. The specific codes entered on the form are the direct inputs to the computer reports.

(3) There are a number of possible computer outputs of the AFM 66-1 system. Some of the available outputs are:

(a) BLIS (Base Level Inquiry System): This output records maintenance activities of a particular base and is normally available only at the base. The BLIS reports are compiled from AFTO Form 349's and are available in various formats.

(b) AFLC D056 Reports: These are computerized reports processed by AFLC that are intended as measures of product performance and maintenance manhours and reflect areas that require improvement. AFLCR 66-15 outlines the different possible reports available. One of the reports (the D056B5006 report) provides equipment historical information on the maintenance actions, manhours, and aborts by month for past six months on every WUC included in the master record. The intent of the report is to show trending and performance data in the areas of failures, maintenance actions, manpower resources expenditure and aborts.

(4) AFTO Form 95 (Significant Historical Data): This optional form can be filled out as a history of maintenance actions taken for a particular LRU every time the LRU comes into the intermediate shop for repair. Actual information displayed can vary from shop to shop. Normally, records are maintained by JCN, WUC and LRU serial number and include actual repair actions/remedies, components replaced, adjustments made, Not Reparable This Station (NRTS) codes, Could Not Duplicate (CND) or bench check retest OK (RETOK) occurrences and test station downtime. When maintained accurately, these records provide a measure of the FAR and indicate which LRUs were removed for BIT fault detection or for other reasons, or both.

(5) USN Standard Maintenance and Material Management System (3M). This system is virtually parallel to the AFM 66-1 system.

(6) The Army Maintenance Management System (TAMMS). The TAMMS also parallels the AFM 66-1 system.

2.4 Testability Effectiveness Measures.(3)

Figure 1 offers a menu of techniques for measuring testability effectiveness. The Figures of Merit provide a point of departure for analyses which might pinpoint precise points for corrective actions. For any particular program, a set of FOM might be selected as best suited to the kinds of data available.

2.5 A Sample Technique for Measurement of BIT Effectiveness.(2)

Three measures of BIT effectiveness (fault detection (FD), fault isolation (FI) and false alarm rate (FAR)) have been offered for operational data evaluation. The matrix (Table 1) shows the four possible BIT conditions that can exist.

TABLE 1. BIT INDICATION MATRIX

BIT INDICATION	TRUE EQUIPMENT STATUS	
	ACTUAL FAILURE	NO FAILURE
→ NO-GO	(correct) a	(false) c
→ GO	(false) b	(correct) d

By interrelating the terms of Table 1, the three measures of BIT effectiveness can be expressed.

a. Fault Detection (FD):

$$\text{BIT FD (\%)} = \frac{a}{a + b} \times 100\%$$

b. False Alarm Rate (FAR):

$$\text{BIT FAR (\%)} = \frac{c}{a + c} \times 100\%$$

c. Fault Isolation (FI):

$$\text{BIT FI (\%)} = \frac{\text{Faults correctly isolated by BIT}}{a} \times 100\%$$

3. Completion Criteria.

This is an open-ended task that may be conducted at one or more intervals throughout the operational life of the system. Testability monitoring may be discontinued at such time as it is established that the testability goals are being satisfactorily achieved in view of the remaining anticipated life of the system.

#	Figure of Merit	Definition	General Model
1	**Fraction of Faults Detected (FFD)**	a) the fraction of all faults detected (or detectable) by BIT/ETE. b) the fraction of all detectable faults detected (or detectable) with BIT/ETE.	$FFD_A = \frac{\text{quantity of faults detected by BIT/ETE } (Q_{BDF})}{\text{quantity of all faults } (Q_F)}$ $FFD_D = \frac{\text{quantity of faults detected by BIT/ETE } (Q_{BDF})}{\text{quantity of faults detected } (Q_{FD})}$
2	**Fraction of False Alarms (FFA)**	the fraction of all BIT/ETE indicated faults which are false alarms. False Alarms are those indications of a fault when an actual fault has not occurred.	$FFA = \frac{\text{quantity of BIT/ETE false alarms } (Q_{FA})}{\text{quantity of all BIT/ETE indicated faults } (Q_{BIF})}$
3	**Fraction of False Status Indications (FFSI)**	The fraction of BIT/ETE fault indications (or lack thereof) which are erroneous.	$FFSI = \frac{\left\{\text{quantity of false alarms } (Q_{FA})\right\} + \left\{\text{quantity of undetected faults } (Q_{UD})\right\}}{\left\{\text{quantity of BIT/ETE indicated faults } (Q_{BIF})\right\} + \left\{\text{quantity of undetected faults } (Q_{UD})\right\}}$
4	**Mean Fault Detection Time (T_{FD})**	The average time it takes for BIT/ETE function to detect and indicate a fault from the time the fault has occurred.	$T_{FD} = \frac{\sum_{i=1}^{Q_{BDF}} (\text{time to detect and indicate the } i^{th} \text{ BIT/ETE detectable fault})}{Q_{BDF}}$
5	**Mean BIT/ETE Running Time (T_B)**	The average active time to perform a BIT/ETE routine. This can be the average for one test, a group of tests, or all tests.	$T_B = \frac{\sum_{i=1}^{NB} (\text{active running time of the } i^{th} \text{ BIT/ETE test routine, } T_{B_i})}{\text{the number of BIT/ETE test routines } (N_B)}$
6	**Frequency of BIT/ETE Executions (F_B)**	The frequency (or cycling rate) at which periodic BIT/ETE tests are executed (This does not apply to BIT/ETE tests that are executed only upon requests).	$F_B = \left\{\text{[the time it takes to execute the complete set of BIT/ETE test routines]} + \text{[the idle time between the execution of the complete set of BIT/ETE test routines]}\right\}^{-1}$
7	**Test Thoroughness (TT)**	The fraction of the equipment/system tested by BIT/ETE relative to the entire equipment/system.	$TT = \frac{\text{(amount of system/equipment tested by BIT/ETE)}}{\text{(amount of system/equipment tested by BIT/ETE)} + \text{(amount of system equipment not tested by BIT/ETE)}}$
8	**Fault Isolation Resolution (FIR(L))**	The fraction of detected faults isolated by BIT/ETE down to an acceptable (specified) minimum number of replaceable items.	$FIR(L) = \frac{\text{quantity of detected faults isolatable to} \le L \text{ RIs with BIT/ETE } (Q_{IL})}{\text{quantity of detected faults } (Q_{FD})}$
9	**Fraction of Faults Isolated (FFI)**	The fraction of faults detected by BIT/ETE, isolated with BIT/ETE to the replacement level specified by the maintenance concept.	$FFI = \frac{\text{quantity of detected faults isolated with BIT/ETE } (Q_{IB})}{\text{quantity of faults detected } (Q_{FD})}$

Figure 1. Figures of Merit (Page 1 of 3)

#	Figure of Merit	Definition	General Model
10	Mean Fault Isolation Time (T_{FI})	The average time to complete the fault isolation process using BIT/ETE.	$T_{FI} = \dfrac{\sum_{i=1}^{Q_{BDF}} (\text{time to isolate the } i^{th} \text{ fault with BIT/ETE})}{Q_{BDF}}$
11	Maintenance Personnel Skill Level (MPSL)	a) the average skill level required to perform corrective maintenance for a system/equipment b) the minimum skill level required to perform corrective maintenance on a system/equipment	Not Applicable
12	BIT/ETE Reliability ($MTBF_{B/E}$)	The probability that the BIT/ETE circuitry will perform its intended function for a specified interval under specified conditions. BIT/ETE circuitry is any hardware that is used for BIT/ETE testing that is not common to the system hardware.	$MTBF_{B/E} = [\lambda_{B/E}]^{-1} = \left\{ \sum_{k=1}^{N_{B/E}} \lambda_{B/E_k} \right\}^{-1}$ $N_{B/E}$ = quantity of BIT/ETE hardware components not common to system hardware
13	BIT/ETE Maintainability ($MTTR_{B/E}$)	The average time to repair a fault in the BIT/ETE hardware.	$MTTR_{B/E} = \dfrac{\sum_{k=1}^{N_{B/E}} \lambda_{B/E_k} M_{CT_k}}{\sum_{k=1}^{N_{B/E}} \lambda_{B/E_k}}$
14	BIT/ETE Availability ($A_{B/E}$)	A measure of the degree to which the BIT/ETE circuitry is in the operable and committable state at the start of a mission, when the mission is called for at an unknown (random) point in time.	$A_{B/E} = \dfrac{MTBF_{B/E}}{MTBF_{B/E} + MTTR_{B/E}}$
15	System Maintainability (MTTR)	The average corrective maintenance time for all system/equipment faults.	$MTTR = \dfrac{\sum_{i=1}^{N} \lambda_i M_{ct_i}}{\sum_{i=1}^{N} \lambda_i}$
16	System Availability (A)	A measure of the degree to which a system/equipment is in the operable and committable state at the start of a mission, when the mission is called for at an unknown (random) point in time.	$A = \dfrac{MTBF}{MTBF + MTTR}$
17	Fraction of False Pulls (FFP)	The fraction of RIs removed from a system, due to the result of a BIT/ETE fault isolation process, that are good RIs (i.e., RIs with no actual failure within it).	$FFP = \dfrac{\text{quantity of good RIs removed } (Q_{GRI})}{\text{quantity of RIs removed } (Q_{RR})}$
18	Fraction of Erroneous Fault Isolation Results (FEFI)	The fraction of BIT/ETE fault isolation results that identify the wrong RI once a fault has been detected.	$FEFI = \dfrac{\text{quantity of erroneous fault isolation results, } (Q_{EFIR})}{\text{quantity of fault isolation results, } (Q_{FIR})}$

Figure 1. Figures of Merit (Page 2 of 3)

BIT/ETE FOM	Analysis Technique	Demonstration Technique
1 Fraction of Faults Detected (FFD)	• can be analyzed by a ratio of occurrence rates (e.g., failure rate)	• can be verified by a binominal distribution or by field data collection (FFD_D only)
2 Fraction of False Alarms (FFA)	• can be analyzed by a ratio of occurrence rates (e.g., failure rate)	• can be verified by field data collection only
3 Fraction of False Status Indications (FFSI)	• can be analyzed by a ratio of occurrence rates (e.g., failure rate)	• can be verified by field data collection only
4 Mean Fault Detection Time (T_{FD})	• can be analyzed by a method similar to MIL-HDBK-472 procedure 2 or RADC-TR-78-169, a failure rate weighted average of times (times determined thru time line analysis)	• can be verified by direct time measurement
5 Mean BIT/ETE Running Time (T_B)	• can be analyzed by time line analysis since there is no randomness in its occurrence	• can be verified by direct time measurement
6 Frequency of BIT/ETE Executions (F_B)	• can be analyzed by time line analysis since there is no randomness in its occurrence	• can be verified by direct eime measurement
7 Test Thoroughness (TT)	• can be analyzed by a ratio of occurrence rates (e.g., failure rate)	• can be verified by direct measurement or the same way FFD is demonstrated, depending on how it is defined
8 Fault Isolation Resolution (FIR(L))	• can be analyzed by a ratio of occurrence rates (e.g., failure rate)	• can be verified by a multinomial distribution or by field data collection
9 Fraction of Faults Isolated (FFI)	• can be analyzed by a ratio of occurrence rates (e.g., failure rate)	• can be verified by a binomial distribution or by field data collection
10 Mean Fault Isolation Time (T_{FI})	• can be analyzed by a method similar to MIL-HDBK-472, procedure 2 or RADC-TR-78-169, a failure rate weighted average of times (times determined thru time line analysis)	• can be verified by techniques similar to MIL-STD-471 or by field data collection
11 Maintenance Personnel Skill Level (MPSL)	• can be analyzed by a weighted average of skill levels if it is defined as an average, otherwise it is strictly determined by measuring the maximum skill level required for each maintenance action	• can be verified by direct measurement or by field data collection
12 BIT/ETE Reliability ($MTBF_{B/E}$)	• can be analyzed using MIL-HDBK-217	• can be verified by using the techniques of MIL-STD-781 or field data collection
13 BIT/ETE Maintainability ($MTTR_{B/E}$)	• can be analyzed using MIL-HDBK-472, RADC-TR-78-169	• can be verified using the techniques of MIL-STD-471 or field data collection
14 BIT/ETE Availability ($A_{B/E}$)	• can be analyzed using current techniques to determine reliability and maintainability (e.g., MIL-HDBK-217, MIL-HDBK-472, etc...)	• can be verified by using the techniques of MIL-STD-781 and MIL-STD-471 or by field data collection
15 System Maintainability (MTTR)	• can be analyzed using MIL-HDBK-472, RADC-TR-78-169	• can be verified by using the techniques of MIL-STD-471 or field data collection
16 System Availability (A)	• can be analyzed using current techniques to determine reliability and maintainability (e.g., MIL-HDBK-217, MIL-HDBK-472, etc...)	• can be verified by using the techniques of MIL-STD-781 and MIL-STD-472 or by field data collection
18 Fraction of Erroneous Fault Isolation Results (FEFI)	• can not be analyzed	• can be verified by a binomial distribution or by field data collection

Figure 1. Figures of Merit (Page 3 of 3)

TESTABILITY TASK COMPENDIUM
Task Reference Number D1B

PHASE: Operation and Support

FUNCTION: Monitor/Evaluate Testability Data

TASK TITLE: Review change proposals for testability

TASK OBJECTIVE: Maintain or improve the integrity and consistency of testability in the design as changes and retrofits are developed.

DESIGN DISCIPLINE TESTABILITY ($\bar{T}$) INTER-RELATIONSHIPS:

$\bar{T}$ INTERFACE	INPUT TO $\bar{T}$	OUTPUT FROM $\bar{T}$
● Change Control Board	● Proposed Changes	● Testability Evaluation
● Design Engineering	● Proposed Changes	● Testability Evaluation

COST TRADEOFF INTER-RELATIONSHIPS: Tradeoff of implementation cost penalties versus future cost saving benefits is an essential step in evaluating testability (or any -ility) candidates for redesign. Also analysis of costs associated with performance improvements is essential to the normal ECP decision processes. Operations and support analysis may involve any of the cost tradeoffs encountered in development phases. Actual or accrued cost data should be accessed wherever available, and parameters used for future cost estimating should be kept current in the light of observations made of actual operations and support.

TASK SYNOPSIS:

1. Task Requirements.

The data collected and evaluated in Task D1A may be analyzed in order to propose modifications to the prime equipment or test equipment as deemed necessary and to provide testability evaluation of all proposed ECPs.

2. Implementation.

Two types of ECPs may arise: performance or other considerations may give rise to change proposals originating in other than testability areas, and the testability monitoring activity may wish to generate changes to improve testability.

2.1 The testability monitoring activity (contractor and/or customer) needs to establish representation on the change review (or control) board (contractor and/or customer) in order to have a voice in approval or modification of proposals to change the system where the changes impact upon testability characteristics.

2.2 The monitoring and analysis of field operations and maintenance data may identify testability problems whose only solution is an ECP, e.g., a change to a test program tape or test adapter. Representation on the change review board(s) is not essential to the generation of ECPs, but the ECP may be facilitated where such representation does exist.

3. Completion Criteria.

The task is a continuous one and exists for as long as there is potential for ECP.

OPERATIONS AND SUPPORT PHASE BIBLIOGRAPHICAL REFERENCES

1. A Study of Testability Standardization for Electronic Systems and Equipment prepared by Testing Technology Office, NSWC and Command & Control Applications Branch, NOSC for NESC, Code 304, pg. 37.

2. BIT/SIT Improvement Project (Phase 1): Evaluation of Selected USAF Aircraft BIT/SIT Systems/Subsystems, ASD-TR-79-5013, pg. 49.

3. BIT/External Test Figures of Merit and Demonstration Techniques, RADC-TR-79-309.

GLOSSARY OF TESTABILITY TERMS

The sources of some definitions are given in parentheses following the definition. The source identifications are:

ATG	Industry ATG Glossary, Report of Industry Ad Hoc ATE Project for the Navy, April 1977
IEEE	IEEE Standard Dictionary of Electrical and Electronics Terms, IEEE Std 100-1972
IEEE/FTC	Interim IEEE Technical Committee on Fault Tolerant Computing Dictionary of Terms.
I/JS	Industry/Joint Services Automatic Test Project Final Report
TR-3826	A Framework for Designing Testability Into Electronic Systems
MIL-STD-721B	Definition of Effectiveness Terms for Reliability, Maintainability Human Factors and Safety
MIL-STD-1309B	Definitions of Terms for Test, Measurement and Diagnostic Equipment.
MIL-STD-2077	Test Program Sets, General Requirements for

A

Active redundancy. That redundancy wherein all redundant items are operating simultaneously, rather than being switched on when needed (IEEE).

Active testing. Closed-loop testing (TR3826). Testing which generates its own test vectors.

Ambiguity group. The group of maintenance replaceable units which may contain faults which result in the same fault signature; also the number of units in such a group (TR-3826).

ATE bit skew. The maximum time difference between the first and last digital pulse arriving at the ATE interface within the same digital test pattern.

ATG node. An ATG system may translate a user's unit under test (UUT) into a logic-equivalent circuit. In general, the translation will not be a one-to-one mapping. The additional nodes thus created are called ATG nodes. It should be pointed out that a logic equivalent usually does not have one-to-one correspondence between the actual hardware of the UUT and its ATG equivalent circuit.

Automatic test equipment (ATE). Equipment that is designed to conduct analysis of functional or static parameters to evaluate the degree of performance degradation and that may be designed to perform fault isolation of unit malfunctions. The decision making, control, or evaluative functions are conducted with a minimum reliance upon human intervention (MIL-STD-1309B).

Availability. A measure of the degree to which an item is in the operable and commitable state at the start of the mission, when the mission is called for at an unknown (random) point in time (MIL-STD-721B). Availability is the probability of a system readiness over a long interval of time (TR3826).

Average isolation time. Average time to follow the worst case logic chain of each diagnostic branch containing three or more unique tests (MIL-STD-2077 (AS)).

B

Back-tracing. A procedure used for automatic test generation which starts from a specified node and traces toward its source or sources until primary inputs are reached.

Binary simulator. A program which establishes a representation of a logic circuit or configuration based upon a computer-directed and/or processed model of the logic circuit or configuration. Node points and output pins of the simulated circuit/configuration are permitted to take on only two values--logical 1 or logical 0-- in sequences and along paths in accord with program rules, in order to derive fault-detection and fault-isolation information for the logic circuit or configuration represented.

BIT false alarms. A false alarm occurs if a component is declared defective and it is found in separate off-line test to be failure free.

BIT - hardware redundant. Hardware redundancy refers to performance replication where functions to be tested are mapped from the operational circuitry to the test circuitry essentially on a 1:1 basis. Such standard hardware redundant BIT methods may be further partitioned into continuous and sampled fault monitoring methods. Representative time continuous approaches are the well-known duplication, triplication, etc. approaches. Non-continuous BIT hardware redundant approaches may be sampled either in time or topologically or both.

BIT - information redundant. Information redundancy refers to the use of coding theory techniques such as those widely used in communications applications. Coding techniques include both systematic and non-systematic approaches. Systematic codes are those where the encoded information can be separated from the data. Non-systematic codes combine the data and the check code so that additional processing is required to separate the two.

BIT performance. A failure for purposes of the specification means that the equipment does not meet the requirements of the procurement specification.

BIT thoroughness. The BIT-detected percent of all failures weighted by failure rates.

Bridge fault. A short fault (TR3826). Faults caused by short circuits between adjacent paths.

Built-in test (BIT). A test approach using BITE or self-test hardware or software to test all or part of the unit under test (UUT) (MIL-STD-1309B).

Built-in test equipment (BITE). Any device which is part of an equipment or system and is used for the express purpose of testing that equipment or system. BITE is an identifiable unit of the equipment or system (MIL-STD-1309B).

C

Casualty. A manifestation of a failure at the system level or major subsystem level such that the system/subsystem is incapable of performing its principal function(s). A casualty is differentiated from a malfunction by the greater seriousness or persistence of its nature (TR3826).

Catastrophic fault, digital. A primary failure in digital circuitry which causes secondary failures (TR3826).

Catastrophic fault, analog. A fault in analog circuitry which causes a sudden change in operating characteristics which results in a complete lack of useful performance. (ATG)

Checkpoint. A place in a routine where a check, or recording of data for restart purposes, is performed (IEEE).

Closed-loop testing. Testing in which the input stimulus is controlled by the equipment output monitor (MIL-STD-1309B).

Comparison tester. The utilization of a known good board ("golden unit") as a means for comparing test results with the unit under test when both are subjected to the same stimuli.

Component-internal fault. A device or component fault which is not a pin fault (TR3826).

Comprehension. The completeness of a test procedure/program with respect to a desired level of fault detection and/or fault isolation. Also, the transparency of an Automatic Test Generation (ATG) program to a specific logic-implementing element or primitive.

Confidence test. A go/no-go test.

Controllability. An attribute of equipment design which defines or describes the degree of test control which may be exercised at internal nodes of interest (TR3826).

CPU. Central Processing Unit (a category of hardware).

Critical failure. A failure which results in a casualty (TR3826).

Critical race. In digital logic, the concurrent change of two or more feedback lines which may result in any one of two or more stable states being entered (TR3826).

D

Deductive simulation. A simulation method in which the failure-free UUT is first simulated and its responses for each stimulus computed. A deductive program is then used to analyze, based on the good UUT responses, what failures will affect the status of which primary output.

Delay fault/delay failure. A failure in a digital device such that switching occurs to the proper level but does so outside of a specified time interval (TR3826).

Dependent fault/dependent failure. A fault which is caused by the failure of an associated item (MIL-STD-721B).

Design fault. A design characteristic of either hardware or software which causes or materially contributes to equipment malfunction independent of the presence of hardware failures (TR3826).

Design for testability (DFT). A design process or characteristic thereof such that deliberate effort is expended to assure that a product may be thoroughly tested with minimum effort, and that high confidence may be ascribed to test results (TR3826).

Diagnostic accuracy. The percentage of failures, based upon the failure population, which are correctly isolated (TR3826).

Diagnostic test. A test designed to perform fault isolation. (TR3826).

Disruptive test. One which destroys (changes) the state of the hardware or software.

Don't-care state. When an input pattern or stimulus is created for a UUT, it is very common that only a portion of the primary inputs will be assigned to specific values (one or zero). Those which are not specifically assigned are said to be the don't-cares.

Dynamic test. A test of one or more of the signal properties or characteristics of the equipment or of any of its constituent items, performed such that the parameter(s) being observed is (are) measured and assessed with respect to a specified time aperture or response (ATG).

E

Early failures. Also birth failures or burn-in failures. Failures during the early period, beginning at some stated time and during which the failure rate of some items is decreasing rapidly (IEEE). A large number of internal circuit failures probably occur in this period due to manufacturing imperfections or design errors (TR3826).

End-to-end run time. The time for a Test Program to determine that a UUT is good (MIL-STD-2077 (AS)).

Equivalent fault. A fault X is equivalent of a fault Y if, and only if, a system containing fault X has the same observable behavior as the same system containing fault Y (IEEE/FTC). Two or more faults, depending on the logic structure, which create the same response for a complete test set. When a single test (one or several stimuli) is applied to a UUT, it is possible that two or more faults may have the same response, but they may react differently for another test. If this is true, the faults are not equivalent.

Error. Any discrepancy between a computed, observed, or measured quantity and the true, specified, or theoretically correct value or condition (IEEE).

Exact match fault dictionary. A fault dictionary whose successful utilization is predicted exclusively upon existence of exact matches of observed fault signatures against predicted fault signatures enumerated in the dictionary (TR3826).

External ATE. ATE which is physically separated from the unit under test when the UUT is in its operational environment (TR3826).

F

Fail-all simulator. Known also as a sequential simulator; all faults simulated one at a time in serial fashion.

Failed machine response (FMR). The output response of a failed UUT when a stimulus is applied.

Fail soft/soft failure. The toleration of the effects of a predetermined number of failures with only partial loss of functional capacity (IEEE/FTC).

Failure. The termination of the ability of an item to perform its required function (IEEE). A failure is the functional manifestation of a fault (TR3826). A malfunction that causes degradation or complete loss of equipment performance (MIL-STD-1309B). The inability of an item to perform within previously specified limits (MIL-STD-721B). A condition of the unit under test which causes the next higher assembly to perform in an unacceptable manner (I/JS).

Failure analysis. The logical, systematic examination of an item or s diagram(s) to identify and analyze the probability, causes, and consequences of potential and real failures (MIL-STD-721B and MIL-STD-1309B).

Failure coverage. An attribute of a test expressed as the percent of failures in the failure population which that test will detect (TR3826).

Failure, dependent. A failure which is caused by the failure of an associated item, distinguished from independent failure (MIL-STD-1309B). One which is caused by the failure of an associated item(s). Not independent (MIL-STD-721B).

Failure, independent. A failure which occurs without being related to t failure of associated items, distinguished from dependent failure (MIL-STD-1309B). One which occurs without being related to the failure of associated items. Not dependent (MIL-STD-721B).

Failure mode. A failure classification (TR3826).

Failure population. The failures which correspond to specified failure modes. This may be used as a basis for the design and evaluation of tests.

Failure, random. Any failure whose occurrence is unpredictable in an absolute sense but which is predictable only in a probabilistic or statistical sense (MIL-STD-721B and MIL-STD-1309B).

Failure rate. The number of failures of an item per unit measure of life (cycles, time, miles, events, etc., as applicable for the item) (MIL-STD-721B).

Failure resolution. A measure of the capability of a test process to perform fault (failure) isolation among replaceable units, generally expressed as N or fewer replaceable units XX% of the time (based upon the failure population) (TR3826).

Failure universe/failure population. The failures which correspond to a selected fault population. This is used as a basis for the design and evaluation of tests (TR3826).

Failure universe. There are many ways to determine the failure universe. A user may specify the universe that suits his purpose. When an ATG system is employed, the system must clearly explain how the universe is defined, and ATG statistics must be based on this specified universe. Percent-detect is meaningless unless the failure universe is clearly defined. Because of theoretical as well as practical limitations, the failure universe is usually defined as the total number of single stuck-at failures of user nodes. Double stuck-at failures and bridge failures are sometimes included, but they have to be specified. When equivalent circuits are used by an ATG, the ATG node failures may be used as the universe and, in this case, separate statistics should be given. Otherwise, the percent-detect may be misleading because a node of the equivalent circuit may not exist in the actual hardware used in the UUT.

False alarm. An indicated fault where no fault exists. (Does not include good items in an ambiguity group.) (MIL-STD-1309B).

False alarm rate. The frequency of occurrence of false alarms (TR3826).

False reject. An item/device incorrectly identified as exhibiting faulty performance due to either a false alarm or an ambiguously isolated fault.

Fault. A physical condition that causes a device, component, or element to fail to perform in a required manner; for example, a short-circuit or a broken wire (IEEE). A degradation in performance due to detuning, maladjustment, misalignment, failure of parts, and so forth (MIL-STD-1309B). The causative failure of a lower level assembly within the unit under test (ultimately a fault may be traced to a physical change of a component or the system) (I/JS).

Fault, analog, catastrophic. A defect or malfunction in an analog component, assembly, or system, causing a sudden change in its operating characteristics which results in a complete lack of useful performance of the device or system.

Fault, analog, out-of-tolerance. A defect or malfunction in an analog component, assembly, or system, in which a performance parameter approximates but falls outside the prescribed upper limit or lower limit for that parameter.

Fault collapse. Since equivalent faults or failures cannot be distinguished without manual intervention, it is desirable to have a preanalysis which permits equivalent faults to be classified or grouped together. For each equivalent class, only one representative can be used for both test generation and fault simulation; the efficiency of the ATG system is thus enhanced. Since the members of all equivalent classes are shown in the fault dictionary, the information is complete and the resolution is the same.

Fault coverage. An attribute of a test or test procedure expressed as the percent of faults of the total fault population which that test or test procedure will detect (TR3826).

Fault detection. A process which discovers or is designed to discover the existence of faults; the act of discovering existence of a fault (TR3826). One or more tests performed to determine if any malfunctions or faults are present in a unit (MIL-STD-1309B).

Fault dictionary. A list of elements where each element consists of a test and all the faults detected by that test (IEEE/FTC). Often only the LRUs which contain the faults are listed (TR3826). A data set or file created by an ATG system. It relates the test stimulus, the responses of the failure-free machine, the various failed-machine responses, the nature of the failure, and the associated replaceable units. The fault dictionary may only show the failure-free UUT responses and indicate which stimuli (pattern) can detect what failure(s) from which primary output (pin) and the associated replaceable units. The difference in data structure results from the different simulation techniques used.

Fault, digital, open. A defect or malfunction in a logic circuit or digital device, resulting from a discontinuity (break) in a signal path such that the output response for any input stimuli becomes indeterminate.

Fault, digital, stuck-at-one. A defect or malfunction in a logic element, circuit, or digital device, such that one or more of its outputs takes on the value of a logical 1 and cannot be changed from this high state regardless of the input stimuli.

Fault, digital, stuck-at-zero. A defect or malfunction in a logic element, circuit, or digital device, such that one or more of its outputs takes on the value of a logical 0 and cannot be changed from this low state regardless of the input stimuli.

Fault dominance. A fault X dominates a fault Y if, and only if, every test for Y is a test for X, where X and Y are two faults which may occur in a system (IEEE/FTC).

Fault indicator. A device which presents a visual display, audible alarm, and so forth, when a failure or marginal condition exists (MIL-STD-1309B).

Fault isolation. Where a fault is known to exist, a process which identifies or is designed to identify the location of that fault within a small number of replaceable units (TR3826). Tests performed to isolate faults within the unit under test (MIL-STD-1309B).

Fault latency time. The duration of time during which an existing fault is undetected; the elapsed time between fault occurrence and fault detection.

Fault localization. Where a fault is known to exist, a process which identifies or is designed to identify the location of that fault within a general area of equipment. Fault localization may be less specific than fault isolation (TR3826).

Fault masking. A fault X masks a fault Y if no test for fault Y is a test for the faults X and Y occurring jointly in a system (IEEE/FTC).

Fault population. The totality of faults which may be incurred by a device (TR3826).

Fault prediction. A process used to predict that some component will be out of tolerance before the next scheduled maintenance period based upon the present measurement of component parameters (TR3826).

Fault resolution. A measure of the capability of a test process to perform failure isolation among replaceable units, generally expressed as "n or fewer replaceable units XX% of the time" (based upon the fault population) (TR3826). A measure of the capability of an ATG to perform failure isolation. In this case, identification of the failed replaceable units (RU's) rather than the failures serves as a measure of the resolution achieved. The fault resolution is considered satisfactory if, after application of a test set to a UUT, the number of RU's which are identified as possibly contributing to a detected fault is equal to or less than a specified number.

Fault signature. An output test vector resulting from the testing of a unit containing one or more faults (TR3826).

Fault simulation. A process which admits prediction or observation of system behavior in the presence of a specified fault without infliction of that fault upon that system. The process demands modeling of either the fault, the system or both (TR3826).

Fault symptom. A measurable or visible abnormality in an equipment parameter (MIL-STD-1309B).

Fault tolerance. The capacity of a computer, subsystem, or program to withstand the effects of internal faults; the number of error-producing faults a computer, subsystem or program can endure before normal functional capability is impared (IEEE/FTC).

FDI. Fault Detection Isolation system. Generic term encompassing all of the above terms.

Field failures. In-service failures, characterized by an absence of burn-in failures (TR3826).

FIT. Fault Isolation Test. Iterative step of BIT that isolates failure to a single component or narrows failure to a group of components.

Functional fault. A fault which can be described by a change in function of some identifiable portion of a system (IEEE/FTC). A failure.

Functional model. When a circuit can be described as a functional unit, it is not necessary to show its logic structure. Instead, a function name or code is given. A UUT can thus be expressed by a functional model which is a network containing several function blocks.

Functional modularity. The splitting of a system into parts or modules based on the function or purpose of those parts (IEEE/FTC).

Functional test. A test which is intended to exercise an identifiable function of a system (IEEE/FTC). The function is often tested independent of the hardware implementation of the function (TR3826).

Functional partitioning. The physical or electrical separation of system elements along interfaces which define and isolate these elements on bases of function or purpose (TR3826).

G

Gate-level model. A modeling technique in which equivalents for high-level logic elements are constructed from basic gating elements.

Global-initialization algorithm. An algorithm is a systematic procedure that can be programmed on a digital computer, which will terminate either when the operation is complete or when it has reached its theoretical limit. For initialization of sequential circuits in ATG systems, the algorithm must not depend on the outputs of the memory elements to be initialized. It should "compute" a sequence of input patterns so that when they are applied to the UUT, the unknown states become known and unique. A global-initialization algorithm is one which is capable of initializing all memory elements of a UUT to its limit.

Go-chain test. A test or a group of tests which evaluate a UUT performance function or parameter.

Good machine response (GMR). The output response of a failure-free UUT when a stimulus is applied.

Go/no-go test. A test designed to yield a "test pass" or "go" indication in the absence of faults in a UUT, and a "test fail" or "no-go" indication in the presence of fault(s) (TR-3826).

Go/no-go. A set of terms (in colloquial useage) referring to the condition or state of operability of a unit which can only have two parameters: (a) go, functioning properly, or (b) no-go, not functioning properly (MIL-STD-1309B).

H

Hard core. That kernel of circuitry in a processor or system which must be functioning properly in order for that processor or system to successfully execute tests of other portions of itself (TR-3826).

Hard core failure. A failure in the hard core logic of a system which inhibits normal self-test of the system (TR-3826).

Hard detect. If the responses of a failed UUT and a good UUT differ by at least one bit, the failure can be definitely detected and is called a hard detect.

Hard fault. A fault which has effectively reached the limit of its effect upon the performance of the next higher assembly (I/JS).

Hazard. In combinational logic, the possible transient changing of an output due to internal delay characteristics. A hazard is harmful if it affects the states of memory devices.

Hazards - static hazard. Is a transition between a pair of adjacent input states which both produce the same output, during which transition an opposite output momentarily occurs. Dynamic Hazard is a transition between a pair of adjacent input states--one of which produces a "1" output and the other of which a "0" output--during which transition it is possible for both a momentary "0" output and a momentary "1" output to occur.

HW. An abbreviation for hardware.

I

Idling. When resource is not being used by the system but is scheduled to be used.

Impossible detect. Failures which cannot be detected by any test (ATG).

Independent fault/independent failure. A fault which occurs without being related to the failure of associated items (MIL-STD-721B).

Initialize. (1) To establish an initial condition or starting state; for example, to set logic elements in a digital circuit or the contents of a storage location to a known state so that subsequent application of digital test patterns will drive the logic elements to another known state; and (2) to set counters, switches, and addresses to zero or other starting values at the beginning of, or at prescribed points in, a computer routine (MIL-STD-1309B).

Input test vector. A test pattern (TR3826).

Interface adapter. A device designed to provide a compatible connection between the unit under test and the test equipment. It may include proper stimuli or loads not contained in the test equipment.

Interface device (ID). The ID shall provide mechanical and electrical connection and signal conditioning, if required, between the ATE and the UUT (MIL-STD-2077(AS)).

Interference testing. A type of on-line testing that requires disruption of the normal operation of the unit under test (see non-interference testing) (MIL-STD-1309B). Off-line testing (TR3826).

Iterative test. A test which must be repeated several times with new values for some of the test parameters each time, provided that all required new test parameter values can be obtained as a function of the iteration number or as a unique transformation on the values used in the immediate preceding iteration. The test to be iterated is defined as a unique test and the number of iterations is the number of additional iterative tests.

Intermittent fault. A temporary fault (IEEE).

Inverted - pyramid/building - block. Descriptive terms characterizing a test or test technique whereby the smallest possible portions of hardware are tested first in the test sequence, and subsequent tests utilize previously verified hardware for execution (TR3826).

L

Latent fault time. The extent or duration of time during which an existing fault is undetected; the elapsed time between fault occurrence and fault detection (TR-3826).

Learn-mode testers. The utilization of random test patterns as stimuli for a circuit to produce a change in state at the output. If no change in state occurs with a given pattern, that pattern is discarded. This process is continued until a series of active patterns are put together to form a test. This type of test generation cannot perform fault isolation.

Line replaceable unit (LRU. A unit which is designated by the plan for maintenance to be removed upon failure from a larger environment (MIL-STD1309B).

LRU. A generic term that may be defined in terms of both Avionic equipment and Ground Systems equipment. For Avionic equipment or systems, it is Line Replaceable Unit. For Ground based equipment or systems, it is Lowest Replaceable Unit. LRU is defined as a unit which is designated by the plan for maintenance to be removed upon failure from a larger entity (equipment, system) in the latter's operational environment (MIL-STD-1309B). A LRU is composed entirely of SRU's.

LSA. Logistic support analysis. A systematic process for defining logistic support requirements, generally in conformance with MIL-STD-1388.

M

Macro block. In some ATG systems, the users can define a logic configuration as a single macro. This is convenient if the logic configuration is used repeatedly many times in a UUT. Moreover, the chance of making errors is greatly reduced when macro blocks are used. Since the ATG system will expand the macro blocks in terms of primitives, there is no limitation on how simple or complex a macro block can be. Also, there is no need to write a special interpretive subroutine for each macro block. Macro blocks are similar to macro instructions for assembly languages. They are also referred to as "open macros" because they are expanded by the ATG system.

Macro function (functional macro). The functional macro is a "closed macro" -- i.e., one not expanded by the ATG system. The functional macro is represented as a "single block" -- i.e., RAM, ROM, counters, shift registers, etc. This single-block representation saves memory space. Special sub-routines are used to compute the input/output relationships. In addition to space savings, this approach can enhance test-generation performance. The functional macro is generally used for MSI and LSI, where fault resolution is relatively unimportant (cf. functional model).

Maintainability. A characteristic of equipment design and installation which is expressed as the probability that an item will be retained in or restored to a specified condition within a given period of time, when the maintenance is performed in accordance with prescribed procedures and resources (MIL-STD-721B).

Maintenance replaceable unit. An LRU (TR-3826).

Malfunction. An error (TR-3826).

Marginal fault. A failure such that some equipment function is impaired or out of tolerance and is of a nature such that catastrophic failure does not occur (TR-3826).

Mean-time-to-repair (MTTR). The total corrective maintenance time divided by the total number of corrective maintenance actions during a given period of time (MIL-STD-721B). (Includes actions due to false alarms.)

Mean time between maintenance (MTBM). The mean of the distribution of time intervals between maintenance action (either preventive, corrective or both) (TR-3826).

Mean time to isolate. The average time required to achieve fault isolation as measured from the time of fault detection to the time of fault isolation (TR-3826).

Micro functional. Cf. primitive model.

Mistake. A human action that produces an unintended result (IEEE).

Module. A pluggable board (card) upon which circuit components are mounted.

Multiple failure. A joint occurrence of two or more single failures (IEEE).

N

Net. The inputs and outputs of logic gates are usually referred to as nodes. A net is a group of nodes connected together.

Non-interference testing. On-line testing (TR-3826). A type of on line testing that may be carried out during normal operation of the unit under test without affecting the operation (see interference testing (MIL-STD-1309B).

O

Observability. An attribute of equipment design which describes the extent to which signals of interest may be observed (TR-3826).

Off-line. (1) Operation of input/output and other devices not under direct control of a device; and (2) Peripheral equipment operated outside of, and not under control of the system, for example, the off-line printer (MIL-STD-1309B).

Off-line ATE. The testing of a unit under test with the unit removed from its normal operational environment.

Off-line testing. Testing of the unit under test removed from its operational environment or its operational equipment. Shop testing (MIL-STD-1309B). Test of a unit under test (UUT) with the unit removed from its normal operational environment (ATG).

Off-line test equipment. Equipment used to perform tests on a UUT with the unit removed from its normal operating environment (ATG).

On-line. Operation of an input/output device as a unit of the system under programmed control of the system (MIL-STD-1309B).

On-line ATE. The testing of a unit under test with automatic test equipment while in its operational environment.

On-line test. Test of a UUT in its operational environment (MIL-STD-1309B).

On-line testing. Testing of the unit under test in its operational environment (see interference testing and non-interference testing (MIL-STD-1309B).

On-line test equipment. Equipment used to perform tests on a UUT while the unit is in its normal operating environment (ATG).

Open fault. A fault caused by an electrical separation of normally electrically connected points (IEEE/FTC).

Output test vector. An ordered set of simultaneously observed output values (TR-3826).

P

Parallel simulation. For digital UUT's, all operations are bit-independent. Since a word of a computer usually consists of many bits, several failures can be simulated simultaneously through a failure-injection mechanism. This is the most popular and accurate method of simulation.

Parametric fault. A fault which causes some parameter for a device to have a value outside its specified range (IEEE/FTC).

Parametric test. The measure of circuit characteristics to ascertain that they fall within specified tolerances (ATG).

Pattern sensitive failure. A component failure, usually internal to the component, whose effect at the component's output pin(s) is dependent upon the input applied (TR-3826).

Passive test. Non-active testing (TR-3826).

Percent detect. Gross percent-detect of user nodes is the total number of user-node failures detected--including possible detects--divided by the failure universe based on user nodes. Gross percent-detect of ATG nodes is the total number of failures detected--including possible detects--divided by the failure universe based on ATG nodes. Net percent-detect is the total number of failures detected positively--not including possible detects--divided by the defined failure universe. Adjusted net percent-detect is the total number of failures detected positively divided by the defined failure universe, which does not include impossible detects.

PID. Peripheral interface device. (A hardware category.)

Pin fault. A fault which is present at a single input or output pin of a component or module (TR-3826).

Possible detect. When a hazard-free UUT is under failure condition or when a UUT is designed with clock pulses and is tested statically, races, hazards, and even oscillations can occur. Consequently, the output response may contain "indeterminate" states. Also, a UUT may not be completely initialized, causing the output response to contain "unknown" states. In both cases, if the unknown or indeterminate bits of a failed UUT response are its only differences from the good UUT response, it is called a possible detect. For binary logic the unknowns or indeterminates can only be either "1" or "0", hence the failure is sometimes detectable.

Primary failure. An independent fault (TR-3826).

Primitives. The basic blocks or operators used by an ATG system. There are several levels of primitives used by different ATG systems. Some systems use combinational gates or Boolean operators only. Other systems include sequential elements such as latches, flip-flops, delay lines, and monostables in their primitive sets. Some systems consider counters, shift registers, ROM, and RAM as primitives. Primitives usually are expanded into equivalent circuits but are handled as single blocks by interpretive subroutines.

Primitive model. When a UUT is described by primitives, the result is a primitive model. It is actually a micro-functional model, which is one level higher than the gate-level model.

R

Races. When a sequential circuit goes from one state to another because of an input change, the input stimulus will usually cause the internal variables to change. During this time, transient conditions can be produced because of the propagation and switching delays of the components. The timing relationships which occur among the internal variables during the transition sequence are affected by these transients and are termed races. Because sequential circuits have memory elements, these races may cause the state transition of the circuit to go wrong momentarily or permanently. A race that can cause a sequential circuit to go to a wrong state permanently is called a critical race and must be avoided.

Random fault/random failure. An intermittent fault whose occurrence is predictable only in a statistical sense.

Readiness. A state of being ready to successfully perform or being in the act of successfully performing a defined mission (TR-3826).

Readiness test. A test specifically designed to determine whether an equipment or system is operationally suitable for a mission (MIL-STD-1309B).

Real-time test A test of one or more of the signal properties or characteristics of an equipment or any of its constituent items, performed such that the parameter(s) being observed is (are) measured and assessed while the equipment is operating at its normal frequency or timing.

Reconfiguration. A repair strategy in which failing components are switched out of operation and replaced by failure-free components (IEEE/FTC).

Recovery. The continuation of system operation with error-free data after an error occurs (IEEE/FTC).

Redundance, redundancy. The introduction of auxiliary elements and components into a system to perform the same function as other elements in the system for the purpose of improving reliability and safety (IEEE). Also, the use of additional components, programs or repeated operations, not normally required by the system to execute its specified tasks, to overcome the effects of failures (IEEE/FTC).

Redundant failure. A failure whose occurrence in a system does not terminate system ability to perform any required function (IEEE/FTC).

Repeatability. A test characteristic such that repeated application of a given set of stimuli to a UUT yields identical results (TR-3826).

Response. The observable reaction of a device to stimulus (TR-3826).

S

Secondary failure. One or more dependent faults (TR3826). Those failures which occur as a result of a previous malfunction depicted as the primary failure.

Self-test. Built-in test (TR3826). A test or series of tests, performed by a device upon itself, which shows whether or not it is operating within designed limits. This includes test programs on computers and automatic test equipment which check out their performance status and readiness (MIL-STD-1309B).

Self-test capability. The ability of a device to check its own circuitry and operation. The degree of self-test is dependent on the ability to fault detect and isolate (MIL-STD-1309B).

Shop non-ambiguity ratio. The shop non-ambiguity ratio is the ratio of the number of detected faults which can be isolated with certainty to the total number of detected faults. A shop non-ambiguity ratio of 1.0 is of course a design goal, but not usually to be economically attained.

Short fault. A fault caused by an electrical connection between normally electrically separated points (IEEE).

SIT (System Integration Test). Built-in system test that is centrally integrated, i.e., BIT data is analyzed through a central computer before Fault Detection/Isolation can be determined.

Soft fault. A fault which has not reached the limit of its effect upon the performance of the next higher assembly (I/JS).

Solid fault. A permanent fault (IEEE).

Specified fault population. A subset of the fault population which is used as the basis for defining the failure universe (TR3826).

SRU (Shop Replaceable Unit). A generic term which includes all the packages within a LRU, which may include circuit boards, chassis, wiring harnesses and piece parts removed at the shop (intermediate or depot) level.

Standby redundance. That redundance wherein the alternative means of performing the function is inoperative until needed and is switched in upon failure of the primary means of performing the function (IEEE).

Static functional test. A test in which measurement is made of the outputs of a unit under test (UUT) after, and only after, these outputs have stabilized with respect to a given input stimulus. Further, the measurement and assessment is made only with respect to the specific, overall action or purpose which the UUT is intended to perform or serve.

Static test. A test in which measurement is made on a UUT after, and only after, these outputs have stabilized with respect to a given input stimulus (ATG modified).

Stimulus. Any physical or electrical input applied to a device intended to produce a measurable response (MIL-STD-1309B).

Stuck fault/stuck failure. A failure in which a digital signal is permanently held in one of its binary states (IEEE).

Supplementary data. Supplementary data consists of information, text, schematics and logic diagrams necessary for analysis of the TPS and UUT in event of a problem or anomaly during the testing process. The amount and content of the supplementary data is contingent upon the capability of the ATE to store and display required information automatically (MIL-STD-2077(AS)).

SW. An abbreviation for software.

Symptom. The manifestation of evidence of a particular failure condition. (TR3826).

T

Ternary simulator. A program which establishes a representation of a logic circuit or configuration based upon a computer-directed and/or processed model of the logic circuit or configuration. Node points and output pins of the simulated circuit/configuration are permitted to take on three values--logical 1, logical 0, or X (unknown state)--in sequences and along paths in accord with program rules, in order to derive fault-detection and fault-isolation information for the logic circuit or configuration represented.

Test. A procedure or action taken to determine under real or simulated conditions the capabilities, limitations, characteristics, effectiveness, reliability, or suitability of a material, device, system, or method (MIL-STD-1309B). A segment of a source program which contains as a minimum, a comparison between a measured value and program defined limits, as well as the branching instructions directing the program to proceed based on the results of the comparison. Generally, it also will contain source statements to apply stimuli and to set up and make measurements. When the stimuli being applied must be calibrated, as when incrementing, or as when RF amplitude calibration is required, each comparison in the incrementing/calibration routine is counted as a test.

Testability. A design characteristic which allows the status (operable, inoperable, or degraded) of a system or any of its subsystems to be confidently determined in a timely fashion (TR-3826). Testability attempts to quantify those attributes of avionic system designs which facilitate detection and isolation of faults that affect system performance. Testability has been defined as "the characteristic of a design which allows the status of a system or any of its subsystems to be confidently determined in a timely fashion." This definition should be expanded to include the concept of isolating and repairing detected faults in a confident and timely fashion so as to minimize repair time. Testability is the unambiguous isolation of a fault to an appropriate level of replaceable equipment such that the proper maintenance is implemented with a minimum expenditure of time and resources. Specification of that appropriate level is a most important consideration, which must be addressed early in the design cycle. (Subtask A-3 Design for Testability, Final Report. U.S. Government Printing Office, 1977, 727-156/1-3, p 35.)

Testability reflects the susceptibility of a PCB to the detection of all faults, to rapid and accurate isolation to the faulty component without ambiguity and to functional test and thereby verification of PCB performance as specified and/or required. (Testability Investigation Attachment IV to Interim Report, Manufacturing Methods and Technology for Digital Fault Isolation for Printed Circuit Boards, Contract No. DAAK40-78-C-0290.)

Testability, comprehensive. An overall testability design characteristic which includes both hardware design and test design.

Testability, inherent. A hardware testability design characteristic which does not include consideration of test stimulus/response data.

Test accuracy ratio (TAR). The ratio of the measurement uncertainty of the UUT to the measurement uncertainty of the ATE. For example, if it is required that a UUT output be accurate to 5% and the ATE accuracy in measuring the parameter is 0.001%, then the TAR is 5000 (MIL-STD-2077(AS)).

Test effectiveness. A measure which reflects the fault coverage and fault resolution provided by a test (TR-3826).

Test generation. The process designing tests or test stimuli (TR-3826).

Test length. The number of tests in a test sequence (TR-3826).

Test pattern. A simultaneous or parallel definition of all the inputs of a system (IEEE/FTC).

Test point. A node within a circuit or system which can be measured or stimulated to facilitate testing (TR-3826).

Test program (TP). The TP contains a coded sequence which, when executed by the ATE, will provide the system a set of instructions sufficient to accomplish the objective of the TP. The objective of the TP is to automatically ascertain the operational readiness condition of the UUT. The TP isolates faults to levels as defined in AR-10 (MIL-STD-2077(AS)).

Test program instruction (TPI). The TPI provides information needed for testing, (E.G., hook-up, probe point locations or other programmed operator intervention) which cannot be conveniently provided by the ATE under control of the TP. Appropriate contents in large part, are dependent on the ATE being used. For example, an ATE with a sophisticated display subsystem could provide diagram and test conveniently and quickly and would need minimal TPIs while on ATE with a slow speed, test oriented, output device would need detailed and extensive TPIs (MIL-STD-2077(AS)).

Test program set. The TPS consists of those peculiar items necessary to test a UUT on an ATE. This includes the electrical, mechanical, instructional and logical decision elements (MIL-STD-2077(AS)).

Test program set integration. The process of debugging the Test Program, Interface Device and Test Program Instruction (MIL-STD-2077(AS)).

Test sequence. A specific order of related tests (MIL-STD-1309B).

Test validation. Actions taken to determine if test responses for a fault-free UUT are in agreement with desired values (TR-3826).

Test verification. Actions taken to assure that a test meets specification of fault coverage and fault resolution (TR-3826).

Topology. When applied to a circuit, topology means the structure of the circuit.

Transient failure. A failure induced by a momentary or temporary external factor such as input power fluctuation, excessive ambient temperature excursion, electromagnetic interference, or by factors internal to a system. A solid fault may cause a transient failure (TR-3826).

Transition count. The number of times an output of a circuit changes state.

Transportability. The ability to convert the output representation of an automatic-test-program generator to a given automatic-test-station target language. This is generally accomplished via a translator.

TRD. An abbreviation for Test Requirements Document.

U

UART. Universal asynchronous receiver/transmitter. (A hardware category.)

Unique test. A test which must be designed individually and is generally executed only once per program run.

Unit under test (UUT). Any system, set, subsystem, assembly, subassembly, and so forth, undergoing testing (MIL-STD-1309B).

Unknown state. Most memory elements used in sequential circuits are bistable devices. When the power is turned on, they can assume either one of the two stable states. Because they are normally designed to have a symmetrical configuration, the initial states become unpredictable and are called unknown states. Unknown states can also be the result of critical races or oscillations.

User node. A node is either an input or an output of a gate or a functional block which may consist of many gates. When a UUT is described by a user to an ATG system, nodes are used to show the interconnections between the gates or the functional blocks. These nodes are called user nodes. Sometimes, a user is allowed to define a macro block which may be used repeatedly in a UUT. The macro blocks will be expanded in terms of gates or smaller functional blocks. Thus, the total number of nodes will be increased, but they are all considered as user nodes.

V

Vector. Input stimuli or patterns are sometimes called input vectors. The responses of a UUT are sometimes called output or response vectors. Mathematically, a vector is a single entity which may contain many components.

W

Wearout failures. Failures during the period during which the failure rate of some items is rapidly increasing due to deterioration processes (IEEE).

Well-behaved failure. A failure whose occurrence produces dependably consistent and predictable symptoms (TR-3826).

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.