RL-TR-97-183
Final Technical Report
October 1997

CHIP TO SYSTEM TESTABILITY

Research Triangle Institute

Michael G. McNamer and Walter W. Weber

Rome Laboratory
Air Force Materiel Command
Rome, New York

RL-TR-97-183 has been reviewed and is approved for publication.

Frank H Born

APPROVED:
FRANK H. BORN
Project Engineer

John J. Bart

FOR THE DIRECTOR:
JOHN J. BART, Chief Scientist
Electromagnetics & Reliability Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** Oct 97	**3. REPORT TYPE AND DATES COVERED** Final Jan 94 - Jun 96

4. TITLE AND SUBTITLE

CHIP TO SYSTEM TESTABILITY

5. FUNDING NUMBERS

C - F30602-94-C-0053
PE - 62702F
PR - 2338
TA - 02
Wu - 6D

6. AUTHOR(S)

Michael G. McNamer and Walter W. Weber

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Research Triangle Institute
3040 Cornwallis Rd.
Research Triangle Park, NC 27709

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Rome Laboratory/ERSR
525 Brooks Rd.
Rome, NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

RL-TR-97-183

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: Frank Born, ERSR, 315-330-3007

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

N/A

13. ABSTRACT *(Maximum 200 words)*

The ultimate objective of the Chip-to-System Testability program was the development of a structured testability implementation methodology which will be used as a basis for a PC-based tool called TESPAD. This tool can be used by development contractors and program offices to establish testability requirements that are verifiable and cost-effective for implementation in electronic systems. Given a set of functional system design requirements, the type of system to be procured (i.e., airborne, ground-based, etc.), and some information about the types of circuits involved in the different levels of the system hierarchy, TESPAD makes use of this structured testability methodology by defining and allocating detailed testability design and validation requirements including testability measures, recommended DFT/BIT methods and structures, data formats and data delivery, and validation/verification procedures. To accomplish this goal, it was necessary to complete a number of predecessor tasks including: 1) Evaluation of positive and negative impacts of testability applied at all levels of the design hierarchy of electronic systems, and 2) Identify chip through system-level test performance that can be achieved (as measured by common testability figures of merit) using current test technology for various types of design technology.

14. SUBJECT TERMS

Testability, System, Requirements, Microcircuit, Digital, Analog

15. NUMBER OF PAGES

228

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Contents

List of Figures

List of Tables

List of Tables (continued)

1. INTRODUCTION

This document details the work of the Chip-to-System Testability program conducted by the Research Triangle Institute and Self-Test Services under contract to Rome Laboratory. Additional details on preliminary tasks are provided in the Interim Report, available from Rome Lab [1].

The ultimate objective of the Chip-to-System Testability program was the development of a structured testability implementation methodology which will be used as a basis for TESPAD, a PC-based tool that can be used by System Program Officers to identify and automatically incorporate testability requirements that are verifiable and cost-effective in the procurement process of electronic systems. Given a set of functional system design requirements, the type of system to be procured (i.e., airborne, ground-based, etc.), and some information about the types of circuits involved in the different levels of the system hierarchy, TESPAD makes use of this structured testability methodology by defining and allocating detailed testability design and validation requirements including testability measures, recommended DFT/BIT methods and structures, data formats and data delivery, and validation/verification procedures.

To accomplish this goal, it was necessary to complete a number of tasks initially, including an evaluation of positive and negative impacts of testability considerations applied at all levels of the design hierarchy of electronic systems and develop evaluation metrics for Design-for-Test (DFT) and Built-In Test (BIT) techniques, structures, data formats, and languages to measure their impact on system life-cycle cost, development schedule, availability, reliability, weight and power requirements, etc. Chip through system-level test performance that can be achieved (quantified using common testability figures of merit) using current test technology for various types of design technology was also evaluated.

1.1. The Need for Testability

As military systems have become increasingly dependent on complex electronic systems, the necessity of testing those systems to ensure reliable performance has increased in a similar fashion. This would not be a concern if testing were a trivial effort; however, it is estimated that testing presently accounts for 30-50% of an electronic manufacturer's cost for a product [2,3]. This percentage, in fact, actually fails to show the true cost of testing for many systems, because it ignores the cost of maintenance efforts. One study of line replaceable module supportability problems (completed as part of the preparation for devising a test approach for the F-22) found that 60% of the dollar-weighted problems can be traced to untestable modules [4].

While the problem of generating test vectors and applying them to combinational digital logic has become tractable in recent years, strictly combinational circuits do not exist in practice, and the problem of testing general sequential circuits remains intractable because an inordinate number of test patterns are required to exhaustively test all possible state configurations [5]. This does not even begin to address the concerns of analog and mixed signal circuitry, for which the definition of what constitutes a test, a good response, and/or a faulty response may be a difficult problem in itself.

Additionally, improvements in test methodologies have been more than offset by increases in system complexity, leading to lower availability and increased maintenance burden [6]. The end result of these limitations is that ineffective testing plagues modern commercial and military systems.

Above the abstract concerns of generating test vectors for a circuit, there are practical considerations of applying those vectors that are driving test costs upward, as well. One issue is the capability and cost of automatic test equipment (ATE). Increased pin counts and/or increases in the size of test pattern sets can make existing ATE obsolete for new systems. New testers, however, represent significant capital investment, which must be amortized over the lifetimes of the systems for which the ATE is employed, adding to the effective testing cost.

The present situation in military applications, however, seems to be even worse than this general scenario, in that observations at an Air Force maintenance depot lead one to believe that there is almost a one-to-one correspondence between the number of specific testers (ATE) and systems to test. Test equipment and associated test software such as this often costs between several hundred thousand and several million dollars, so that this scenario represents an enormous expense. Moreover, much of the tester equipment being used today is outdated, but because the test software is specific to that tester and that system, the tester must be retained. Upgrade is out of the question since the cost of test software is often in the million-dollar range for a single application, and in a tester-per-system environment, the cost of the tester and test software cannot be distributed. However, without upgrade, the ATE being used to determine system failures is itself failing, and in many cases, the diagnostic tests used to evaluate the ATE are not sufficient to determine that there is a problem [7].

An additional practical problem encountered for military systems results from the test software itself. Warner-Robins ALC maintenance personnel have indicated that, in many cases, the test programs used may meet some level of testability requirement, such as fault isolation to an ambiguity group of size five or less, but that the information may simply not be true (i.e., for the fault isolation example, an ambiguity group of size five may be indicated, but the actual fault may not exist within those five components). This is estimated to occur as much as 50% of the time. This scenario represents a classic case of specifications for testability not being sufficiently verified. The previously cited F-22 study found that 75% of the remaining dollar-weighted problems for LRU supportability (30% overall) are due to non-robustness of the TPS software (maintainability or re-targetability problems).

Another consideration, equally important as the cost of generating and applying a test, is the cost of insufficient testing. Problems at one level of a design are magnified at higher levels, and are more costly to fix once integrated. For instance, a component reject ratio may appear to be reasonably low, but a board composed of those components may have an unacceptably high reject ratio because individual defect probabilities are multiplied for an assembly. For example, if the probability of selecting a good component is 0.995 (5000 ppm reject ratio), and a board is made up of 50 of these components, the probability of a good board is $0.995^{50} = 0.78$. Hence, 22% of these boards will be defective. Conversely, a manufacturing failure rate of 1% for a printed circuit board (PCB) consisting of 100 ICs requires that the probability of selecting a good IC be 0.9999, or 100 ppm.

One scenario in field operation and maintenance that may arise from either insufficient testing or ineffective testing is a Can Not Duplicate (CND) or Re-Test OK (RTOK) condition, in which some level of field diagnostic indicates a failure in the system that is impossible to recreate when subsequent testing is performed. As might be expected, CND events are capable of consuming a huge amount of test resources (maintenance personnel, ATE time, etc.) if they occur with any frequency. It has been estimated by Northrop that "a 4% CND/RTOK rate accounts for 30% of the maintenance manpower [for a particular system]" [4]. Unfortunately, CND events are also not rare (in fact, a 4% rate would be considered extremely good). American Airlines, employing maintenance strategies similar to the Air Force, reports that 40 to 60% of the LRUs returned to a shop for maintenance have CND conditions, typically increasing as the complexity (although not necessarily size) of the LRU increases [8].

1.2. Hierarchical Design and Test Methodology

One means of improving the testability of systems is to adopt a structured, hierarchical design methodology, and a corresponding hierarchical test methodology, as depicted in Figure 1-1. Using such a strategy, the test requirements for a particular level of hierarchy can be viewed as the test requirements for each module at the next lower level of hierarchy, and the interconnect logic for these modules. For example, suppose that a circuit board is designed with a structured methodology so that feedback between levels of the hierarchy is eliminated and is composed of five application specific integrated circuits (ASICs) and some interconnect logic. Testing of this board requires first testing each of the ASICs, then using board-level Design-For-Testability (DFT)/Built-In Test (BIT) structures to facilitate testing of the interconnect logic between the ASICs. There is no reason to reapply vectors to test the ASICs when testing the board, other than to verify the pin connections for each applied ASIC. Similar examples can be made for subsystems composed of boards and systems composed of subsystems. A study detailed in [9] found that a 25% reduction in defects per unit for sub-modules corresponds to a 10-15% reduction in module test cost. The key is to ensure that a means is provided to isolate logic to a particular level of hierarchy during test, and at each level of the hierarchy to test the logic specific to that particular level.

The key to implementing and obtaining the full benefits of a hierarchical test methodology, however, is the adoption of "concurrent engineering" principles. Efficient use of the hierarchical test model requires early specification of test requirements and planning for the validation of those requirements (along with the other initial design specifications) and early planning for the use of structured DFT and BIT to enhance the testability of the design. Subsequently, the requirements for testability and plans for DFT/BIT application must be refined as the level of detail of the design increases, through to actual implementation. Finally, verification of the initial and refined requirements must be carried out to ensure that the testability goals are met. In short, testability must be treated as a key design parameter, along with speed of execution, reliability, development cost, etc.

In the sections that follow, we will examine different aspects of this concurrent engineering strategy for testability improvement. We will review our previous work, reported in the Interim

Report [1], on the use of TFOMs, which can be applied to the specification of test requirements and the verification that those requirements are met. We will investigate the general benefits and penalties of applying DFT/BIT to circuit and system designs, and evaluate the trade-offs of specific DFT and BIT techniques that might be recommended up front in a design cycle and then refined as the level of detail increases. We will present a strategy for establishing a set of verifiable testability measures to ensure that the initial requirements specified for testability can be demonstrated to have been achieved and for tailoring detailed testability technique recommendations to enable the achievement of those requirements. Finally, we will present the automation of this strategy in the TESPAD, Testability Specifications Advisor, software tool.

1.3. Report Organization

Section 2 presents a review of the Interim Report [1], which studies the identification and evaluation of the cost/benefit impact of testability and the feasibility of meeting various levels of system testability requirements. Section 3 examines the trade-off relationships indicative of the cost effectiveness of test techniques. The analyses covered in Sections 2 and 3 are incorporated in the development of a methodology for tailoring detailed testability requirement to system procurement, which is presented in Section 4. The implementation of the structured testability methodology, in the form of a PC-based software tool, is detailed in Section 5. Section 6 highlights some of the observations of the technical staff on the difficulties encountered in the development of this work. Section 7 details the conclusions of this effort.

2. SUMMARY OF PRELIMINARY TASKS

To develop a structured methodology for testability implementation, it is first necessary to understand testability requirement allocation and verification. Work to that end was presented in the Interim Report [1], and is summarized below.

This work was divided into two tasks, which are explained in the next two sections.

2.1. Identify and Evaluate the Cost/Benefit Impacts of Testability (Task 1)

The overall goal of Task 1 is to identify and evaluate the positive and negative impacts of testability in a diverse range of military electronic systems such as avionics and ground-based systems. As a minimum, the impact of testability on system life-cycle cost, acquisition cost, development schedule, support equipment, power requirements, reliability, maintainability, sparing levels, circuit complexity, weight, area, and throughput needs to be determined.

An overview of testability metric, or testability figure of merit (TFOM), application, and the advantages and limitations of utilizing TFOM programs in the design flow of a system were examined. A similar analysis was performed for Design-for-Test (DFT) and Built-In Test (BIT) techniques, detailing the high-level advantages and penalties of including DFT/BIT structures in a system design. Data was presented from DFT/BIT success stories, and more importantly, a number of case studies of TFOM and/or DFT/BIT application that allow for tradeoff analysis of the costs and benefits of including testability measurement or structures in a system design.

The evaluation of testability measures that are used/calculated by common TFOM programs or DFT design rule audit programs was also presented. The measures were evaluated for their usefulness as specifications, their feasibility of accurate measurement, their ambiguity, and their feasibility of verification. Using this analysis, a set of recommended measures that can be verified at acceptance test was defined from among the initial set of possible measures from TFOM tools, and possible measurement strategies for each were presented. This is important because specifying testability measures is a costly endeavor, with potentially zero return on investment if the measures cannot be verified before the system is put into service.

Also, an evaluation of specific DFT and BIT techniques was presented, based on a number of testability performance and life-cycle cost criteria. The criteria set for which the techniques are evaluated consists of:

1. Area Overhead
2. I/O Overhead
3. Power Overhead
4. Computational Expense of Design Translation
5. Performance Degradation
6. Test Run Time
7. Stuck-at Fault Coverage
8. Delay, Open, and Bridging Fault Coverage
9. Impact on Reliability, Availability, and Maintainability
10. Applicability to Various Circuit Structures
11. Compatibility with Other DFT/BIT Structures
12. Compatibility with IEEE Standard 1149.1 and Other Test Buses
13. ATE/Diagnostics Accessibility
14. Life-Cycle Usefulness

This analysis was conducted for DFT/BIT strategies pertaining to component, board, and system level designs.

Implications of Task 1

The use of testability measures and the inclusion of testability structures in a design can greatly reduce the difficulty and cost of test and maintenance. However, in few (if any) cases does the use of testability structures or testability measures come without a cost. DFT and/or BIT structures almost always add to the development and operational cost of a system. Moreover, the use of testability structures or measurement techniques to target a particular aspect of testability (such as fault detection) will often have a negative impact on some other aspect (or aspects) of testability (such as false alarm rate). These trade-offs, however, can almost always be balanced so that the net effect of increasing testability or including testable structures is significantly positive, as demonstrated by the testability success stories and case-study trade-off analyses documented in the Interim Report.

One of the keys to maximizing the benefits of including testability considerations in a design is the judicious use of testability measures and structures. Testability measures that are specified for a system must be verifiable at a relatively early point in the design cycle (such as acceptance testing) to reap the full benefit of the testability analysis. Later verification (or lack thereof) may be of use for future endeavors (as empirical data), but is useless in influencing the test costs for that system because the cost of system changes at that point will be too great to warrant alterations. For this reason, we have made a significant effort to demonstrate the identification/development of testability measures that are verifiable and may be used for quantifying the required testing capability.

Similarly, the cost-benefit trade-offs for DFT and BIT structures that are included in a system can only be maximized if system considerations that relate to test and maintenance costs (such as application scenario, relative importance of particular fault models, etc.) are used to match candidate test structures to system applications. For this reason, we have devoted a substantial part of the Interim Report to the analysis of DFT and BIT structures that might be used in systems, and their particular strengths and weaknesses in relation to system test and maintenance parameters. The results of both the testability measure research and DFT and BIT structure analyses are included in Appendices I-A and I-B.

2.2. Determine Feasibility of Meeting Various Levels of System Testability Requirements (Task 2)

The goal of Task 2 was to determine the feasibility of meeting various levels of system testability requirements in terms of fault detection and isolation through analysis of factors that may affect the fulfillment of such requirements.

Documentation of the work of this task presents a study of the levels of commonly specified TFOMs that are practical to achieve for modern electronics systems. The objective was to determine practical levels of TFOMs that can then be used in combination with the results of Task 1 to guide the development of an automated means of generating testability specifications.

The set of TFOMs studied included:

Fraction of Faults Detected (FFD)
Fraction of Faults Isolated (FFI)/Fault Isolation Resolution (FIR(L))
Fraction of False Alarms (FFA)/False Alarm Rate (FAR)
Mean Detection Time (T_D)/ Mean Isolation Time (T_I)

An evaluation of the each of these TFOMs was then detailed. Tools and techniques were investigated for each TFOM, and the following information was evaluated:

1. Specific TFOMs used to measure a particular design characteristic (e.g., FFD for fault detection coverage)
2. Accepted definition(s) for each TFOM
3. Technical assumptions for the measurement of each TFOM (fault models, error models, etc.)
4. Theoretical basis and tools/techniques for the measurement of each TFOM
5. Capabilities of the tools and techniques used to measure each TFOM
6. Accuracy of the analysis provided by the tools and techniques

7. Applicability of the TFOM(s) and tools/techniques across different circuit technologies
8. Applicability of the TFOM(s) and tools/techniques across different levels of system hierarchy
9. Applicability of the TFOM(s) and tools/techniques across different levels of abstraction
10. Applicability of the TFOM(s) and tools/techniques across different test phases
11. Cost of measurement for each TFOM using identified tools and techniques

In many ways, this work echoes the previous work of Task 1; however, the in-depth analysis of tools and techniques performed here allows us to better evaluate the penalties of measurement for recommended TFOMs and to better evaluate the limitations (e.g., on the complexity of designs that can be measured, the level of hierarchy for which the TFOMs can be measured, the level of abstraction at which the TFOM can be measured, etc.) of even "highly recommended" TFOMs, due to measurement constraints.

Based on the above information and data collected from the literature, the penalties incurred by a design due to the specification of certain levels of TFOMs were explored. In particular, attempts were made to relate the cost of achieving a particular level of a TFOM (e.g., FFD) to the level achieved. Cost may be associated with extra design time, extra design cost (through BIT insertion, for instance), extra test time, etc.

An analysis of levels of TFOMs for fault detection, fault isolation, and false alarms that are practical to achieve and measure was presented. This data is crucial to the development of an automated system for the generation of testability specifications because without some knowledge of the level of testing capability that is practically achieved, the specifications generated may be meaningless.

While the definition of testability figures of merit (TFOMs) that are feasible to measure (verifiable TFOMs) have been established, questions remain regarding what levels of testability (as measured by TFOMs) can be feasibly achieved, with respect to both design and measurement costs. Testability requirements that are to be included in system specifications need to be feasible under the cost constraints for a design, and they must also be consistent with other design requirements.

In the current state of procurement, there is little to indicate that any consideration is given to whether or not testability specifications are achievable. Rather, in many instances, it appears that the levels of testability specified are obtained through the use of random selection with predetermined, but arbitrary, upper and lower bound.

Hence, we investigated the TFOMs that are typically specified in a procurement, including:

1. Fault Detection Coverage (FFD)
2. Fault Isolation Coverage (FFI, FIR(L), FI_P)
3. False Alarm Rate (FFA, FAR)
4. Mean Test Time (T_D, T_I)

to determine how they fit into testability specifications for a system and what levels can be achieved cost effectively. A summary of these investigations is included in Appendix II.

Implications of Task 2

TFOMs have been identified and analyzed for digital, analog, and mixed signal circuits. All TFOMs discussed, except the false alarm rate TFOM, have been found to be potentially capable of being used for specifying valid, verifiable requirements, provided that the assumptions made in their prediction, verification, and measurement are noted beforehand for possible limitations on the scope of interpretation (for instance, 95% FFD at the board level for a bit-flip fault model should not be taken to guarantee 95% FFD for all defect and failure modes). The false alarm rate TFOM for analog circuits lacks any practical means for prediction, and even questionable means for verification, so it is deemed impractical and hence, invalid, as TFOM for requirements specification.

Much R&D work still needs to be carried out in the industry to develop additional, practical methods of analyzing the testability of analog circuits and to develop tools for supporting such analyses. Particular attention should be given to developing a better understanding and definition of the false alarm phenomena, so that prediction and verification methods and tools can be developed.

3. DEVELOP TRADEOFF RELATIONSHIPS INDICATIVE OF THE COST EFFECTIVENESS OF TEST TECHNIQUES (TASK 3)

3.1. Task Description

The goal for this task was to develop tradeoff relationships that identify the cost effectiveness of the application of the testing techniques, structures, languages, and data formats examined in Tasks 1 and 2 relative to system application and life cycle. These tradeoff relationships will be developed using the performance analysis results and the evaluation of positive and negative impact of testability insertion results obtained in the previous tasks [1].

This task was necessary because it allowed us to develop tradeoff relationships that relate testability improvement techniques to their cost of implementation, and therefore offered a means for specifying the appropriate DFT and BIT techniques so that a system design can achieve its testability requirements at a minimum cost. To be effectively used in an automated software system, it was also vital to have these tradeoff relationships developed at different levels of the design abstraction and for different types of electronics systems, implying the full use of the results of Tasks 1 and 2.

The tradeoff relationships were defined primarily by performing literature searches [10-25] and, in some cases, by communicating with field maintenance personnel. Several test cases involving the techniques being examined were found in the literature, however the amount of available information was still limited. Difficulties were encountered when requesting military documentation for use as test cases because most of the information is either classified or simply not recorded.

The tradeoff relationships for the set of previously identified BIT and DFT techniques were established using a set of evaluation criteria similar to that of Task 1. Quantitative scores were then assigned to the techniques for each of the criteria, and the totals for each technique were stored in a technique scoreboard database.

To fully gauge the performance improvement gained through, and the implementation cost of, the use of testability techniques, the benefits and penalties associated with each technique must be evaluated in a consistent manner. To this end, a set of criteria was previously developed with which to evaluate the cost effectiveness of each technique. Fourteen fundamental criteria were developed as follows:

1. Area overhead
2. I/O overhead
3. Power overhead
4. Computational expense of design translation
5. Performance degradation
6. Test run time
7. Stuck-at fault coverage
8. Delay/open/bridging fault coverage
9. Impact on availability, reliability, maintainability
10. Applicability to various circuit structures
11. Compatibility with other DFT/BIT structures
12. Compatibility with IEEE Standard 1149.1 and other test buses
13. ATE/diagnostics accessibility
14. Life cycle usefulness

The candidate DFT and BIT techniques were grouped into digital and analog technique categories to aid in the evaluation of the techniques. The categories are as follows:

Digital Techniques (10 categories, 76 techniques)

1. Structured Chip-level BIT (12 techniques)

2. Structured Chip-level DFT (5 techniques)
3. Ad hoc DFT (13 techniques)
4. RAM BIT (11 techniques)
5. ROM BIT (4 techniques)
6. PLA BIT (7 techniques)
7. Coding (7 techniques)
8. Fault Tolerance (7 techniques)
9. Board level testing (6 techniques)
10. System level testing (4 techniques)

Analog Techniques (3 categories, 83 techniques)

1. BIT (48 techniques)
2. Structured DFT (7 techniques)
3. Ad hoc DFT (28 techniques)

To ensure that the criteria used to evaluate the techniques are appropriate to the techniques in each category, the evaluation criteria were tailored to each particular category using the fourteen fundamental criteria above as a basis, deleting inappropriate criteria, and augmenting with additional criteria, as appropriate. For instance, the power overhead and I/O overhead criteria were not used to evaluate the techniques in the coding category as these design costs are generally unaffected by the implementation of coding-type testability techniques (Cyclic codes, Hamming codes, etc.), while error detection and error correction capabilities are benefits of the coding techniques that must be considered to fully appreciate the enhanced performance afforded by the addition of testability.

The criteria were evaluated for their effect on the cost effectiveness of the implementation of the techniques over each stage of the life cycle by assigning a weighting factor to each criteria over each of three life cycle phases: research and development, manufacturing, and field. This life-cycle weighting is important because it reflects the changing of the relative importance of the evaluation criteria as the system moves through the different phases in its life cycle. For example, maintenance technicians who service fielded equipment do not really care how difficult it was to automate the self-testing technique in a circuit while it greatly concerns the development team. In the field, the main concern is how quickly and accurately a failure can be diagnosed and how difficult it is to interface with a failed system. The life cycle weighting of the criteria allows the relative comparisons of techniques to account for this life cycle adjustment.

An example of a technique scoreboard at this point of the evaluation is shown in Table 3.1. The total score, T, can be compared across the technique category to relate the cost effectiveness of the techniques in that category. Note that this score cannot necessarily be compared across categories as the evaluation criteria may be different, resulting in different maximum and actual scores.

Table 3.1. Sample Technique Evaluation

Category	R&D	Man	Field	Total
1	a	b	c	d=a+b+c
2	e	f	g	h=e+f+g
◆	◆	◆	◆	◆
◆	◆	◆	◆	◆
◆	◆	◆	◆	◆
				T=d+h+...

The life cycle weighting consideration for cost effectiveness greatly depends on the application environment in which the technique is to be implemented. For example, area, I/O, and power allocations are at a premium in space-deployed applications. Therefore, each of these criteria are weighted more heavily for space applications. Also, in space-deployed applications, field maintenance is not possible; therefore, ATE/diagnostics accessibility is weighted much more lightly in the field life-cycle evaluation.

Further, each technique was weighted versus one of three application environment scenarios: ground applications, airborne applications, and space applications. This step was necessary to account for the variation in effectiveness for particular DFT and BIT techniques relative to their application environment and other requirements implied by that environment. For instance, BILBO is an effective BIT technique for digital components; however, its high logic overhead cost makes it unattractive for space applications that require minimal power and weight. Similarly, BIT techniques in general become more cost-effective than general DFT as the application becomes more critical, due to the necessity of detecting faults within the application environment implied by the more critical scenarios.

Table 3.2 reflects a technique scoreboard following full evaluation.

Table 3.2. Sample Technique Evaluation

Category	Ground				Air				Space			
	R&D	Man	Field	Total	R&D	Man	Field	Total	R&D	Man	Field	Total
1	a_g	b_g	c_g	$d_g=a_g+b_g+c_g$	a_a	b_a	c_a	$d_a=a_a+b_a+c_a$	a_s	b_s	c_s	$d_s=a_s+b_s+c_s$
2	e_g	f_g	g_g	$h_g=e_g+f_g+g_g$								
•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•
				$T_g=d_g+h_g+..$				$T_a=d_a+h_a+..$				$T_s=d_s+h_s+..$

Once the evaluation process was established, the techniques were evaluated by assigning a score of high, medium, or low on each of the criteria relevant to that technique category. A score of high gives the technique the full numerical score of the life cycle weighting factor for that criterion. A score of medium gives the technique one-half of the maximum score, and a score of low gives the technique a score of zero for that criterion. For each application environment, (ground, air, and space) the scores were totalled over all life-cycle phases and all relevant criteria for each technique. Although the scores assignments are ultimately a subjective analysis, they reflect the combined design and test experience of RTI and Self Test Services personnel. Ultimately, the scores can be altered as desired to augment the TESPAD tool developed using this data.

The scoreboards containing the cost effectiveness tradeoff relationships are contained in Appendix III.

The structured test methodology described in Section 4 uses the results obtained from this task for tailoring testability requirements that include specific testing techniques to system procurement.

4. METHODOLOGY DEVELOPMENT FOR TAILORING DETAILED TESTABILITY REQUIREMENTS TO SYSTEM PROCUREMENT (TASK 4)

4.1 Task Description

The primary objective of this program is the development of a methodology for a software tool, TESPAD, which is capable of tailoring detailed testability requirements that include appropriate testing techniques, structures, data formats, languages, and approaches for system and subsystem procurement. The objective of the methodology is to produce testability requirements that are suitable for inclusion in a Statement of Work for different types of electronic system procurement.

To effect this development, it is necessary to combine all of the previous information developed about feasible testability measures that may be achieved through the use of identified DFT/BIT techniques, structures, and approaches with known cost effectiveness, for the development of verifiable testability requirements tailored to specific system procurements. The developed methodology is the formalism required for relating all aspects of testability requirements to other system constraints in a way that these requirements are feasible to achieve and verify.

4.2. Structured Testability Methodology

Our approach must consider the entire testability hierarchy for a system at all modes of testing and reporting test results. A conceptual hierarchy for testability requirements allocation and DFT/BIT implementation/verification is shown in Figure 4.1. Given a hierarchical configuration imposed by testability requirements, it becomes evident that the effectiveness of DFT/BIT in reporting the system status and/or fault detection/isolation results for all DFT/BIT structures depends largely on the proper interfacing of hardware components in the system configuration. The use of standard interfaces, including standards such as IEEE 1149.1, 1149.5, and P1149.4, is encouraged in the suggested methodology to ensure the proper interfacing which will allow proper test initiation and reporting of test results.

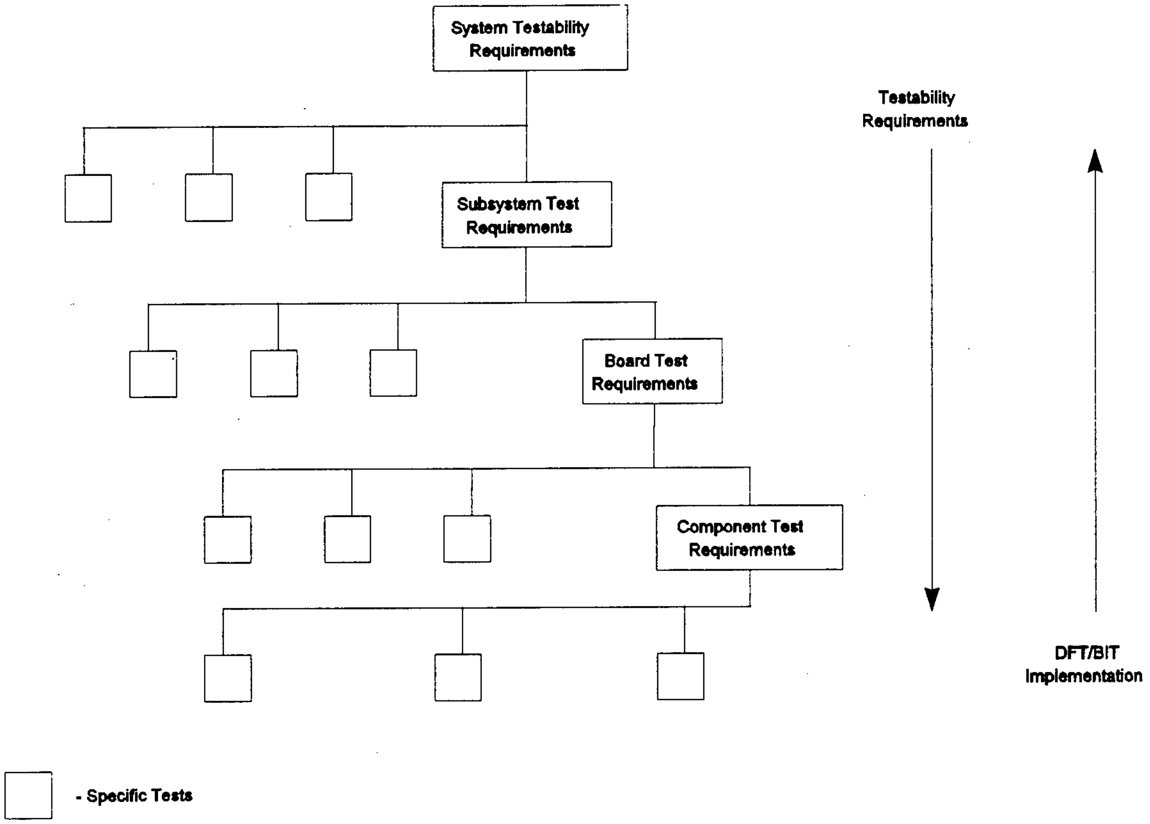


Figure 4.1. Structured Testability Implementation

As shown in Figure 4.1, testability requirements are allocated top-down in the hierarchy and DFT/BIT techniques are assigned bottom-up. The testability requirements are defined based on mission requirement data such as availability, failure rate, etc., which are defined by the system's application and historical data. The consistency of the testability requirements throughout the hierarchy is crucial to maintain attainable specifications; this check on the consistency of testability requirements is demanded by the methodology before any DFT/BIT techniques are assigned.

The testability requirements allocation is followed by recommendations for DFT/BIT techniques that can be used to achieve the stated testability requirements in a cost-effective manner. General recommendations are first produced for the entire system to establish a structured testability approach. Detailed recommendations are then generated within that structured approach for each unit in the system. The detailed recommendations are reported in an order of testability need, determined by factors such as fanout, feedback loops, and failure rate. The DFT/BIT technique recommendations process is explained in detail in Section 4.2.3.

4.2.1. Testability Requirements Allocation

An evaluation of testability measures and requirements, performed as an initial task of this project, was presented in the Interim Report [1]. Several Testability Figures of Merit (TFOMs) were analyzed to determine if they were specifiable, verifiable, and measurable (without excessive measurement cost). Three TFOMs that met those requirements and are widely accepted in the world of military design and test are Fraction of Faults Detected (FFD), Fraction of Faults Isolated (FFI), and Estimated Ambiguity Group Size (E{AG}). These measures were incorporated into the methodology to serve as the testability requirements. The allocation of these requirements is performed top-down throughout the system by dynamically calculating the requirements as the system is defined, which necessarily occurs from top to bottom. The TFOMs are calculated from high-level performance and maintenance specifications through the use of nine models - four for FFD, three for FFI, and two for E{AG}. A check on the consistency of the values for FFD and FFI is performed at the completion of the system description input to ensure the attainability of the values. The consistency check is explained in Section 4.2.2.

The dominant factors in maintenance costs are cost of detection, cost of isolation, size of isolation ambiguity group, and false alarms. The first three factors are covered by the TFOMs calculated in this methodology, FFD, FFI, and E{AG}; the fourth factor, false alarms, is not treated in this methodology because of the lack of a good model (other than empirical CND rates) for FAR/FFA (False Alarm Rates/ Fraction of False Alarms).

4.2.1.1. FFD Models

Four models may be used to calculate the Fraction of Faults Detected requirement, depending on the availability of user inputs and system configuration. The selection of the model to be used to specify the requirement is described after the discussions of each of the four models.

FFD Model 1

The first model for FFD is based on a model for reject ratio from [26],

$$RR = (1-Y)(1-FC) \tag{4-1}$$

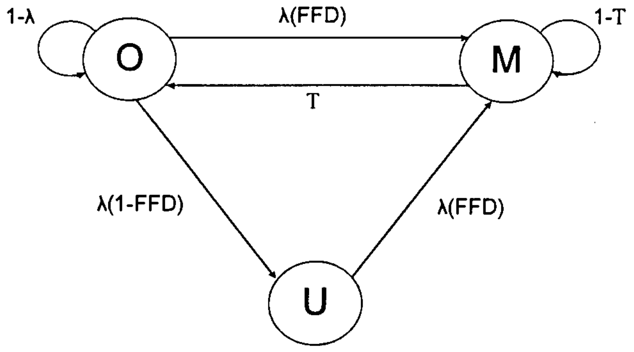
where, RR = reject ratio, Y = expected yield, and FC = fault coverage. Substituting FFD_1 for FC, and solving for FFD_1, gives

$$FFD_1 = 1 - \frac{RR}{(1-Y)} \tag{4-2}$$

We have chosen not to use the classic Williams-Brown equation for reject ratio because of indications that model is primarily valid for very high (95%+) fault coverage levels. Since the input data used in Model 1, reject ratio and expected yield, is specific to production test measurements, this model is primarily applicable to "development only" procurements. However, under the assumption that fault detection in the field should be at least as good as in factory test, Model 1 can also be effectively applied for any system, if the necessary data is available. Since this is the most accurate of the four FFD models, it is included in the testability requirements allocation process even though the probability of the user having access to the necessary input data is low.

FFD Model 2

The second model for FFD uses reliability relations and Markov modeling under the assumption of perfect fault isolation. This assumption is valid in this case because we want to specifically target the necessary level of FFD to meet the maintenance requirements. Using the assumption of perfect fault isolation, which allows the assumption of perfect repair, the state (Operational, Maintenance, or Unknown) of a system can be represented by the following Markov model,



The availability of the system can be calculated as the steady state of P(O), defined as the percentage of time the system operates correctly over the total operating time:

$$P(O) = Availability = A = \frac{\mu FFD}{\mu + \lambda FFD} \tag{4-3}$$

We can then substitute μ = repair rate = $MTTR^{-1}$, and λ = failure rate = $MTTF^{-1}$. Solving for FFD gives

$$FFD_2 = \frac{(A)(MTTF)}{MTTF - (A)(MTTR)} \tag{4-4}$$

The necessary input for this model, availability, mean time to repair (MTTR), and failure rate, is comprised entirely of maintenance requirement levels, and hence should be highly available.

This model is applicable to systems for which the fault detection figure of merit must have a field emphasis (the general military maintenance case), since it is based on predicted, or required, field reliability measures. This is important because this is the dominant paradigm in procurement specifications, since the majority of cost is incurred in the field.

FFD Model 3

The third model is taken from the derivation of a combined value for Fractional Isolability, FI_P, in [27]. This equation for E{AG} is used in [27] to estimate the expected number of sub-element removals in the process of fault isolation and is based on a single failure assumption.

$$\frac{1}{E\{AG\}} = \frac{1}{FFD_C + (1-FFD_C)(n)} \tag{4-5}$$

where n = number of child units and FFD_C is the composite FFD for those child units, as calculated by

$$FFD_C = \sum_{i=1}^{n} FFD_i \left(\frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \right) \tag{4-6}$$

Using estimates for E{AG} according to hierarchy level,

$$E\{AG\}_a = \begin{cases} 1, \textit{ if level} = \textit{system or subsystem} \\ 3, \textit{ if level} = \textit{board} \\ 5, \textit{ if level} = \textit{component} \end{cases} \tag{4-7}$$

an FFD estimate can be calculated, given only the hierarchical connectivity (number of children) of a unit, by

$$FFD_3 = \frac{E\{AG\}_a - n}{1 - n} \tag{4-8}$$

FFD Model 4

The fourth model for FFD uses historical data for COTS (Commercial Off-the-Shelf) parts.

$$FFD_4 = FFD_{COTS} \tag{4-9}$$

Values for COTS testability requirements (FFD and FFI) are maintained in a database for use by the tool. The user will be responsible for the initial entry (and subsequent revision, as necessary) of an FFD value or estimate for COTS parts, which will be recalled upon each use of that COTS part.

Selecting an FFD model

The selection of the FFD model depends upon the available input information and the comparison of the values produced by each of the models. For COTS parts, Model 4 is always used since it is taken directly from historical data. For non-COTS parts, calculation of TFOMs is performed dynamically by the tool when values for availability, MTTR, failure rate, and expected yield are available as inputs, with model selection determined by the following criteria:

- If values for yield and reject ratio are provided, FFD is assigned the lesser value from Models 1 and 2. The lesser of these two values is chosen because of the pessimism of achieving these values.
- If no value is provided for reject ratio and multiple children are defined for the current unit, E{AG} is calculated using the equation listed in the discussion of Model 3 using

FFD_2 in place of FFD_C, and this value is compared with the $E\{AG\}_a$ estimate based on hierarchy level described in the discussion of Model 3. If the difference of the two values is greater than 50%, Model 3 is used and the user is notified of the use of the estimate for $E\{AG\}_a$ and the subsequent use of the estimate in the FFD model. If the difference is less than 50%, Model 2 is used for FFD. Basically, there is no justification for the selection of the comparison threshold value of 50% - it was picked "by feel". This is an aspect that might be altered with empirical data.

- If no value is provided for reject ratio and no children or a single child are defined for the current unit, Model 2 is used as the default value for which the probability of obtaining the necessary input data is highest.

The selection of the FFD models is shown in the TFOM Calculations Flow Diagram, Figure 4.2.

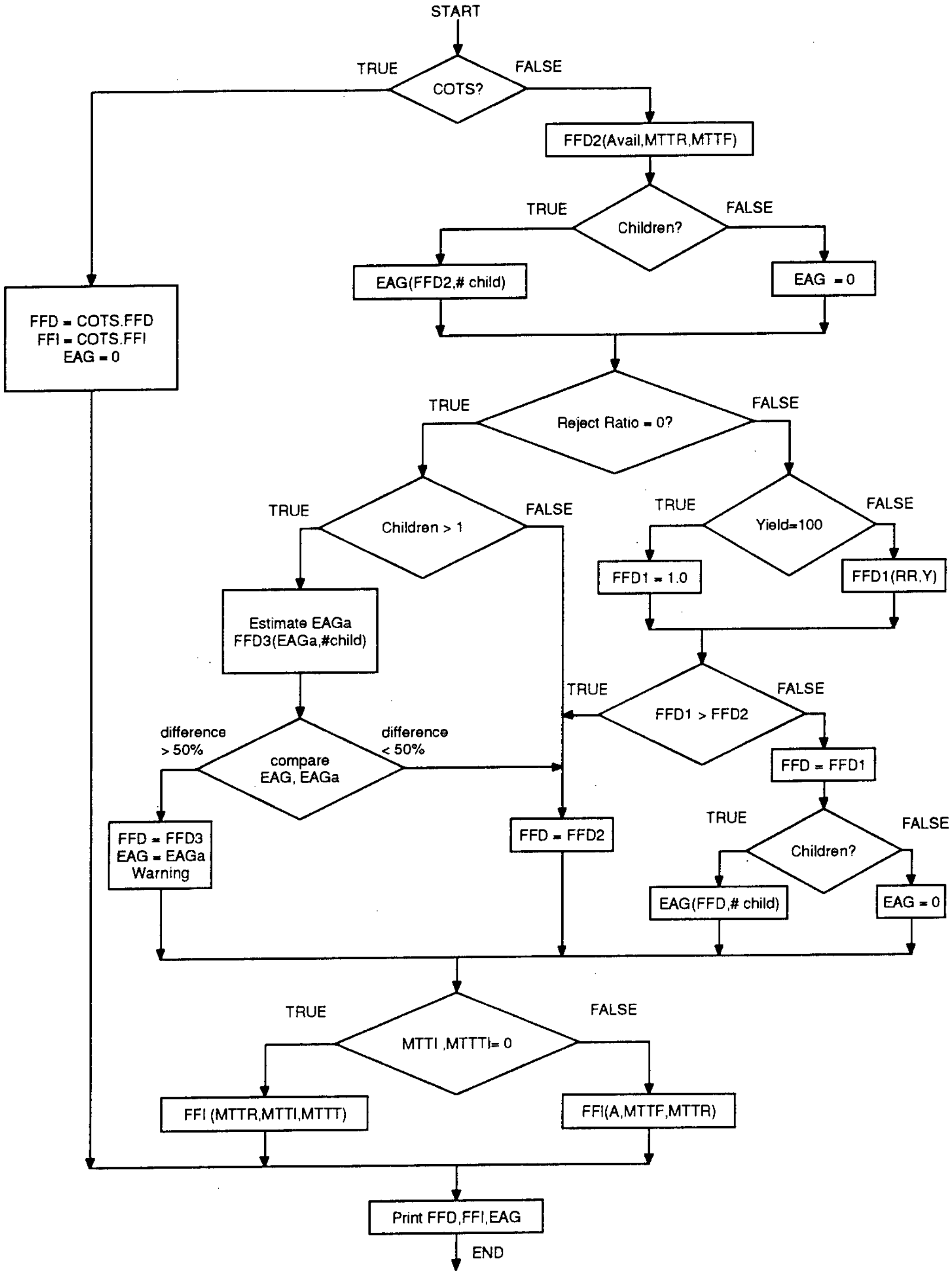


Figure 4.2. TFOM Calculations Flow Diagram

4.2.1.2. FFI Models

Three models may be used to calculate the Fraction of Faults Isolated requirement, again depending on user input availability and system configuration. The selection of the model to be used to specify the requirement is described after the discussions of each of the three models.

FFI Model 1

Working from basic reliability relations, we approximate mean time to repair (MTTR) as

$$\begin{aligned} MTTR &= MTTI + MTTRP \\ &= FFI(MTTI_A) + (1-FFI)MTTI_T + MTTRP \end{aligned} \qquad (4\text{-}10)$$

where, MTTRP = mean time to replace, $MTTI_A$ = mean time to isolate using automated means, and $MTTI_T$ = mean time to isolate using troubleshooting means. Assuming that MTTRP is negligible compared to MTTR, $MTTI_A$, and $MTTI_T$ results in the first FFI model,

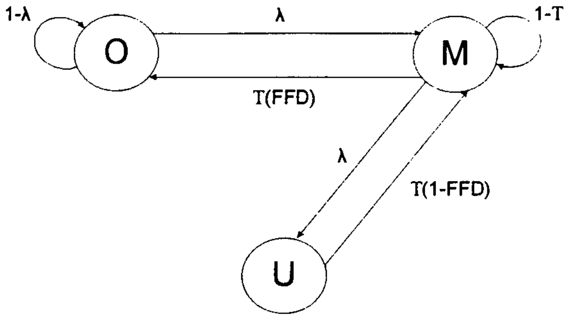
$$FFI_1 = \frac{MTTR - MTTI_T}{MTTI_A - MTTI_T} \qquad (4\text{-}11)$$

This model requires predicted or historical field values for mean isolation times, and is therefore only applicable to units for which that information is available. Default values for each of these required inputs can be saved in the MTTR database based on the application description and unit type fields. The MTTR database is described in Section 5.2.2.2.

FFI Model 2

Similar to FFD Model 2, the second FFI model uses reliability equations and Markov modeling under the assumption of perfect fault detection. This assumption is valid in this case because we want to specifically target the necessary level of FFI to meet the maintenance requirements. Using the assumption of perfect fault detection, which allows the assumption of perfect repair, the state

(Operational, Maintenance, or Unknown) of a system can be represented by the following Markov model.



The availability of the system can be calculated as the steady state of P(O), defined as the percentage of time the system operates correctly over the total operating time:

$$P(O) = Availability = A = \frac{\mu FFI}{\mu + \lambda} \tag{4-12}$$

We can then substitute μ = repair rate = $MTTR^{-1}$, and λ = failure rate = $MTTF^{-1}$. Solving for FFI gives Model 2:

$$FFI_2 = \frac{(A)(MTTR + MTTF)}{MTTF} \tag{4-13}$$

This model assumes perfect fault detection capability; therefore, FFI is defined as the isolated fraction of all faults, not just detected faults. This is the default model for non-COTS parts in the absence of sufficient data for Model 1. The necessary input data for this model, availability, mean time to repair (MTTR), and failure rate, consists entirely of maintenance requirement levels and therefore should be highly available.

FFI Model 3

The third model for FFI uses historical data for COTS (Commercial Off-the-Shelf) parts.

$$FFI_3 - FFI_{COTS} \qquad (4\text{-}14)$$

Values for COTS testability requirements (FFD and FFI) are maintained in a database for use by the tool. The user will be responsible for the initial entry (and subsequent revision, as necessary) of an FFI value or estimate for COTS parts, which will be recalled upon each use of that COTS part.

Selecting an FFI Model

The selection of the appropriate FFI model depends upon the available input information. Model 3 is used for all COTS parts since it is taken directly from historical data. For non-COTS parts, Model 1 is given priority since it is the most accurate; but, in the absence of the necessary input data for Model 1, Model 2 is used as the default whose input data should be readily available. The selection of the FFI models is shown in the TFOM Calculations Flow Diagram, Figure 4-2.

4.2.1.3. E{AG} Models

Two models are used to calculate the Estimated Ambiguity Group Size requirement.

E{AG} Model 1

The first model is taken from the derivation of a combined value for Fractional Isolability, FI_P, in [27]. This equation for E{AG} is used in [27] to estimate the expected number of sub-element removals in the process of fault isolation and is based on a single failure assumption.

$$\frac{1}{E\{AG\}} = \frac{1}{FFD_C + (1 - FFD_C)(n)} \tag{4-15}$$

where n = number of child units and FFD_C is the composite FFD for those child units, as calculated by

$$FFD_C = \sum_{i=1}^{n} FFD_i \left(\frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \right) \tag{4-16}$$

Therefore, we estimate E{AG} as:

$$E\{AG\} = FFD + (1 - FFD)(n) \tag{4-17}$$

The value for FFD is based on the appropriate model for FFD as outlined above.

E{AG} Model 2

The second model for estimated ambiguity group size is based entirely on hierarchy level.

$$E\{AG\}_a = \begin{cases} 1, \text{ if level} = \text{system or subsystem} \\ 3, \text{ if level} = \text{board} \\ 5, \text{ if level} = \text{component} \end{cases} \tag{4-18}$$

This model is used in conjunction with FFD Model 3.

Selecting an E{AG} Model

Model 1 is used as the default model for the Estimated Ambiguity Group Size requirement. However, if no information is given for the reject ratio and multiple children are defined, the estimate from Model 2 will be compared with the value calculated with Model 1. If the difference of the two values for E{AG} is greater than 50%, the estimate for Model 2 will be used as the E{AG} requirement. Otherwise, Model 1 is still used.

4.2.2. TFOM Consistency Check

After calculating FFD and FFI values top-down for each unit in a system, a consistency check is performed on those values to ensure that the values are actually attainable. The attainability of the specified requirements is important both for acceptance testing and for logistics support which is based on the assumption of the attainment of the testability requirements. The consistency check consists of a bottom-up check that is performed on each parent unit (any unit that has child units) in the system until the entire system's TFOM values are verified. In the example hierarchy shown in Figure 4.3, the consistency check would be performed as follows: the FFD and FFI values for module 4 would be checked against the values for modules 1, 2, and 3; the values for module 6 would then be checked against the values for modules 4 and 5.

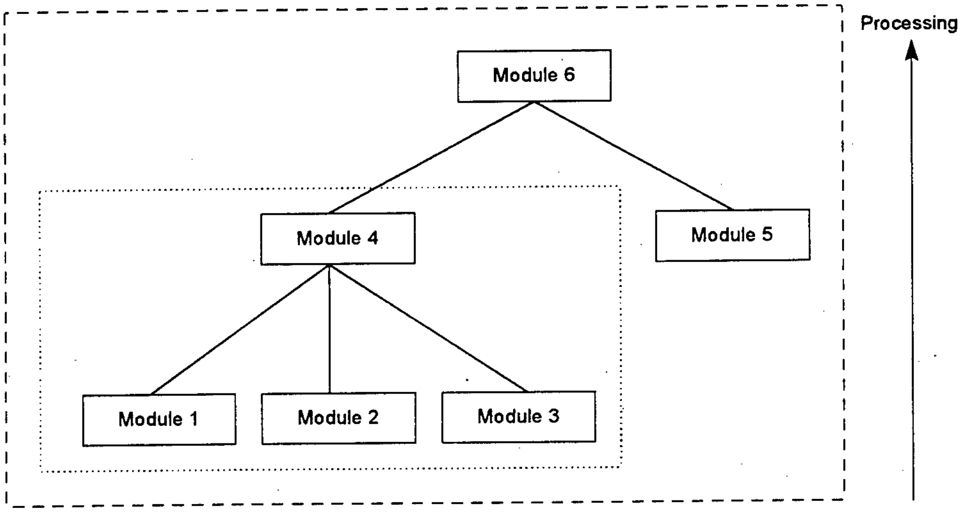


Figure 4.3. Consistency Check Example

The checking calculation for FFD is as follows for each parent unit:

$$FFD_C = \sum_{i=1}^{n} FFD_i \left(\frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \right) \tag{4-19}$$

where λ_i = failure rate of child unit i, and FFD_i = FFD of child unit i. Each of the summations is performed over the number of child units, n. If the check value, FFD_C, for the parent unit exceeds the value calculated from the models, the check value replaces the model-calculated value. If, on the other hand, the model-calculated value exceeds the check value by more than 2%, an inconsistency is indicated because the model-calculated value cannot be attained for the parent unit using the child units as specified. An identical check is performed on FFI values.

4.2.3. DFT/BIT Technique Recommendations

The allocation of consistent testability requirements through TFOM calculation is only one part of the overall testability implementation process; a way of attaining those requirements must also be provided. The structured testability methodology we have developed incorporates the recommendation of Design-For-Test and Built-In Test techniques which provide for the attainment of the testability requirements allocated through the TFOM calculations. The recommendations are presented as suggestions, with several (up to five) technique options at each point of implementation, instead of as rigid specifications and structures to allow the designer freedom to choose the most appropriate technique for the specific implementation.

The recommendations made by TESPAD are based on the cost effectiveness measures developed for the DFT/BIT techniques in Task 3. General recommendations are produced for the system to give an overall strategy for DFT/BIT implementation for the system. Detailed recommendations are listed for each unit in the system based on hierarchy level, circuit type (analog/digital/mixed-signal), criticality of performance, criticality of fault detection, component description (if applicable), sequentiality, and the application environment.

The general technique recommendations for the system are summarized as *test architectures*. These test architectures are used to build a generic framework which will ensure a consistent test methodology throughout the system hierarchy. Four test architectures, TA1, TA2, TA3, and TA4, which become the framework for the bottom-up implementation of DFT/BIT, are defined and used in this methodology:

TA1 BIST and boundary scan (1149.1/1149.4) on digital and analog/mixed-signal components, with fault tolerance at the board level and coding at the component level connected through a hierarchy of test buses (1149.1 (for digital)/1149.4 (for analog/mixed-signal) and 1149.5/TI ASP/National Scan-Bridge) and maintenance controllers.

TA2 BIST and boundary scan (1149.1/1149.4) on digital and analog/mixed-signal components connected through a hierarchy of test buses (1149.1 (for digital)/1149.4 (for analog/mixed-signal) and 1149.5/TI ASP/National Scan-Bridge) and maintenance controllers.

TA3 Structured DFT and boundary-scan (1149.1/1149.4) on digital and analog/mixed-signal components connected through a hierarchy of test buses (1149.1 (for digital)/1149.4 (for analog/mixed-signal) and 1149.5/TI ASP/National Scan-Bridge) and maintenance controllers.

TA4 Ad hoc DFT on digital and analog/mixed-signal components and boards connected through a hierarchy of test buses (1149.1 (for digital)/1149.4 (for analog/mixed-signal) and 1149.5/TI ASP/National Scan-Bridge) and maintenance controllers.

Because each of the four test architectures assign bus-type structures at the system and subsystem level, several test architectures can be used in the same system, each communicating through the test buses at the system and subsystem level. Therefore, the test architectures are assigned at the board and component level only. At the system and subsystem levels, test bus structures are assigned which allow the board and component level test architectures to communicate.

Several sub-hierarchies, to which one of the four test architectures is assigned, are defined within the overall system hierarchy. The overall system hierarchy is partitioned into sub-hierarchies by ignoring all subsystem and system level units. All remaining intact hierarchies, which consist only of board and component level units, are classified as sub-hierarchies. One of the four test architectures is assigned to each of these sub-hierarchies.

The test architecture assignment is based on criticality of fault detection, criticality of performance, and the application environment. Each unit in each sub-hierarchy is evaluated to determine the appropriate test architecture for that unit. Then the sub-hierarchy is traversed to determine the highest priority test architecture assigned in that sub-hierarchy. (Lower test architecture numbers indicate higher priority.) That test architecture is then assigned to each unit in the sub-hierarchy. Once test architectures are assigned to the system, the selection of particular techniques for detailed recommendations is simplified since the type of technique is defined (DFT, BIT, or ad hoc).

In Task 3, the DFT/BIT techniques examined in Tasks 1 and 2 were evaluated for cost effectiveness by developing tradeoff relationships between testability improvement and cost of implementation. Each DFT/BIT technique has been evaluated over a number of different scoring criteria in order to establish an objective selection method in the recommendation process. Techniques were assigned a score of high, medium, or low on such criteria as: area overhead, power overhead, performance degradation, fault coverage, life cycle usefulness, etc. Techniques were evaluated only on those criteria pertinent to their particular technique category; e.g., the power overhead and input/output (I/O) overhead criteria were not used to evaluate coding techniques. Each criterion was then scaled according to its effect at different life-cycle phases

(R&D, manufacturing, and field) and also according to the target application environment (ground, air, or space). The resulting numerical scores were totalled for each application environment and used in the selection of appropriate techniques within each category for technique recommendations.

Technique Recommendation Process

Test architecture assignment is only the first step in the technique recommendation process. After the test architectures have been assigned, the system topology is examined to determine intra-layer feedback loops and points of high fanout. This information allows the recommendation of techniques to be prioritized by unit so that the most critical testability implementation can be identified.

DFT/BIT recommendations are prioritized through the following process:

Step 1. The test architecture assignments are made for all board- and component-level units according to criticality of fault detection, criticality of performance, and application environment within the sub-hierarchies as described above.

Step 2. A fanout check is performed to analyze the system interconnections and identify those units whose fanout (number of outgoing unidirectional connections) exceeds the estimated ambiguity group size of the parent. Units identified with excessive fanout are targeted for DFT/BIT implementation in an attempt to improve the isolation of faults within the established ambiguity groups.

Step 3. All intra-level feedback loops in the system are identified by examining the unidirectional interconnections defined for the system. Units in feedback loops are targeted for DFT/BIT implementation in attempt to improve the isolation of detected faults within the feedback loops.

Step 4. All units are prioritized using the results of the fanout check, the feedback loop check, and each unit's failure rate. The priorities are as follows (each group ranked by failure rate):

1. Unit's fanout exceeds limits and unit is included in a feedback loop
2. Unit with highest failure rate in each feedback loop, but fanout does not exceed limits and, therefore, does not meet condition (1)
3. Unit's fanout exceeds limits but is not included in a feedback loop
4. Unit is included in feedback loop, but not included in conditions (1) or (2)
5. All other units

Failure rates are used to rank the units within the established priority groups in order to target the units most likely to fail. Priorities 1-3 are recommended as "Essential to achieve testability specifications" since they contain characteristics which lead to problems in fault isolation. Priorities 4-5, which increase the testability of the system, yet are not deemed critical, are recommended "To ensure testability specifications; balance versus cost."

Step 5. Recommendations are produced in the order established in step 4. The technique scoreboards for the appropriate technique category, as established in step 1, are accessed to determine the best DFT/BIT techniques. The recommendations are pulled from the appropriate technique database according to the level of assembly, circuit type (analog/digital/mixed), test architecture, and, if applicable, component description. Once the scoreboard has been selected, all techniques contained in that database are ranked by their score for the particular application environment (ground, air, or space). All techniques meeting the following two criteria are included in the recommendations:

1. Techniques whose database score ranks in the top five of all considered techniques
2. Techniques whose database score is at least 75% of the highest score

In the order given by the prioritizing step, techniques are assigned to units in the system by accessing technique scoreboards. The detailed recommendations are compiled and combined with the general recommendations for the overall system to form the complete DFT/BIT testability recommendations for the system.

5. DEVELOP, TEST AND DEMONSTRATE A PC-BASED SOFTWARE TOOL (TASK 5)

5.1 Task Description

The goal of this task is to develop a PC-based, Windows 3.1 tool which automates the structured testability methodology developed in Task 4 and produces detailed testability requirements suitable for inclusion in system specifications for contractual (or sub-contractural) purposes. Given a set of functional system design requirements, the type of system to be procured (i.e., airborne, ground-based, etc.), and some information about the types of circuits involved in the different levels of the system hierarchy, the tool makes use of this structured testability methodology by defining and allocating detailed testability design and validation requirements including testability measures, recommended DFT/BIT methods and structures, data formats and data delivery, and validation/verification procedures.

5.2. The TESPAD Tool

The Testability Specifications Advisor (TESPAD) tool was developed in response to Task 5 to implement the methodology of Task 4 in a Windows-based, graphical environment. TESPAD produces detailed testability requirements data for electronics systems based on information on a system's hierarchical structure, reliability requirements, mission and performance related requirements, and application.

On-line help provides information on the operation of TESPAD, including a textual tutorial which describes a program flow typical of the use of the tool. On-line help can be accessed at several points in the program, including the main menu and each of the dialogs used in the program.

5.2.1. Input Requirements

As shown in Figure 5.1, there are two primary inputs to TESPAD. The first required input is a description of the system hierarchy, including item descriptions and performance specifications.

The item description includes relevant circuit characteristics, level of assembly, maintenance level (LRU/SRU), mission and life-cycle cost requirements, and the item's reliance on off-the-shelf parts. The mission and life-cycle requirements of the item are specified by the following parameters: application environment, availability requirements, mean time to repair requirements, requirements for fraction of field rejects, criticality of performance, and criticality of fault detection. The second required input is a description of the system's interconnections, which are analyzed by the program to locate feedback loops and points of excessive fanout. All input is entered through a graphical interface and then saved by the program in project database files.

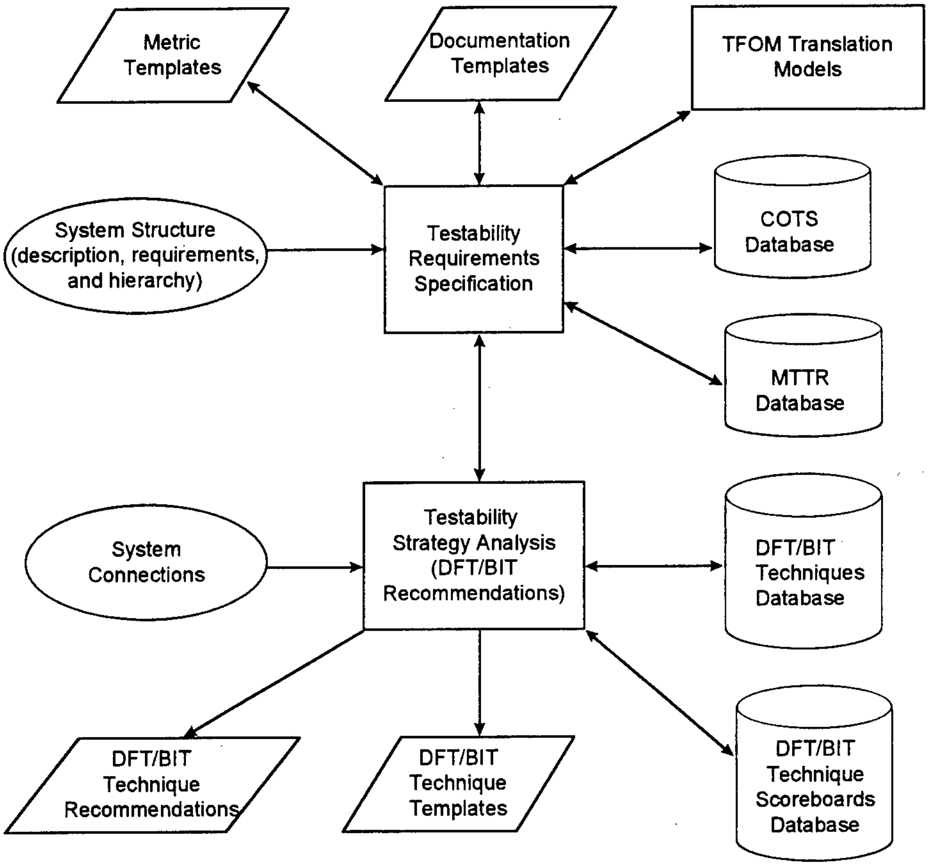


Figure 5.1. Block Diagram of TESPAD

5.2.1.1. Project Database Files

Input for each system is contained in project database files in Borland Paradox format. The tool provides all necessary database interactions through the Borland Database Engine, which is embedded in the tool. Knowledge of database concepts or operation is not required to use TESPAD, nor is Paradox or the Database Engine a requirement.

A project database is named by the user at the time it is created; each specific system is represented by its own project database. The project databases contain the following fields on each unit defined in the system: unit name, level of assembly, parent unit, child units, connections to other units, application environment, application description, unit type, maintenance level (LRU/SRU), circuit type (analog/digital/mixed-signal), technology (bipolar/CMOS/mixed), component description, sequentiality, criticality of fault detection, criticality of performance, availability, MTTR, failure rate, expected yield, reject ratio, COTS status, COTS description, and TFOMs and other data calculated and used by the program. Each of these fields is explained in greater detail in Section 5.2.1.2.1.

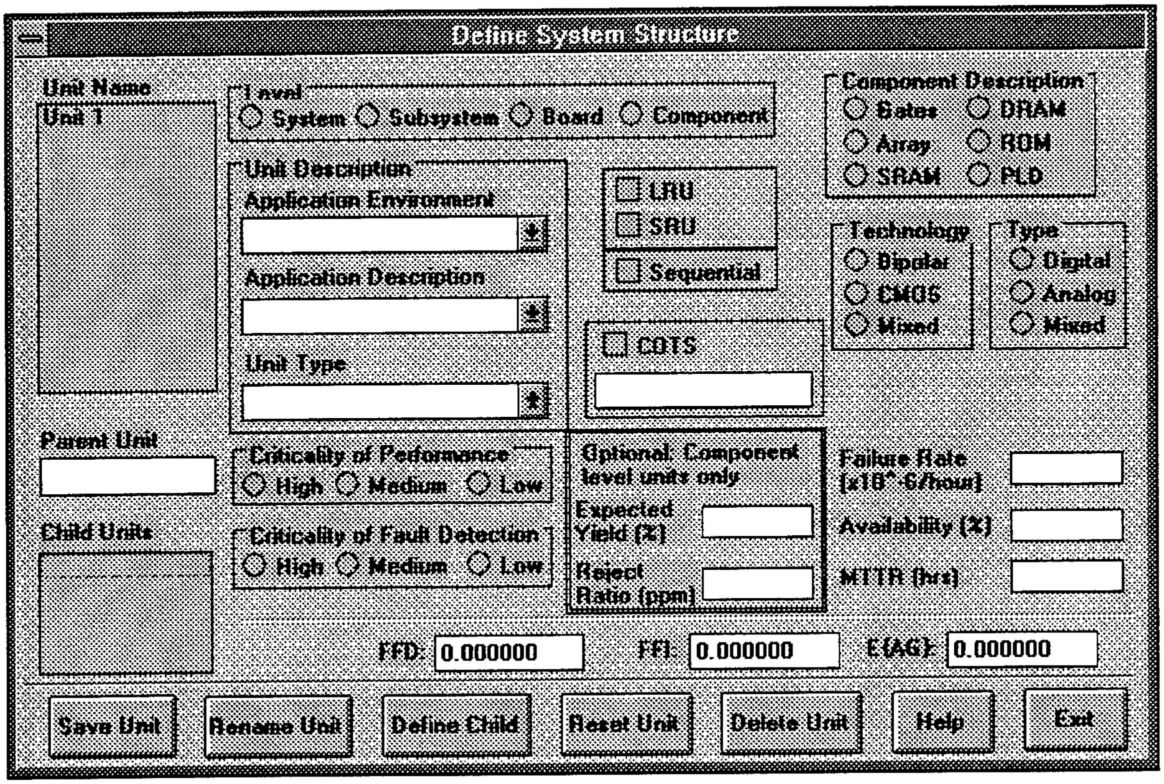
5.2.1.2. Input Dialogs

Two input dialogs perform the graphical interface for project data collection. The System Structure Dialog allows the user to enter information on the system hierarchy, application description, and mission and performance related requirements. The System Connections Dialog allows the user to enter information on the system interconnections.

5.2.1.2.1. System Structure Dialog

The System Structure Dialog, shown in Figure 5.2, allows the user to describe the system hierarchy, application information, mission and performance requirement data, and individual unit description data. The System Structure Dialog also performs all TFOM calculations, as detailed in sections 4.2 and 5.2.3.

The system hierarchy is described in terms of "units". A unit describes any entity in the system, e.g., component, board, module, etc. The top-most unit in the system is named by the user when a new project is created; all other units in the project will be defined as child units of the top-most unit. Usually, the entire system is designated as the top-most unit so the rest of the system can be described by child units of that top-most unit. (The system unit becomes the "parent unit" of the subsystem units in this scheme.) Selection of the level of assembly (system, subsystem, board, or component) affects DFT/BIT technique recommendations and identification of intra-layer feedback loops.



Unit Name: Unit 1

FFD: 0.000000

FFI: 0.000000

E(AG): 0.000000

Figure 5.2. System Structure Dialog

The system's application description, e.g., "Ground Mobile", is entered through the Unit Description comboboxes. Each unit can then be described further using the Application

Description and Unit Type comboboxes. Example entries for application description include: communications, computer, controls/display, electronic warfare, etc. Example entries for unit type include: alarm, antenna, coder/decoder, computer, control, etc. Default values for MTTT, MTTI_A, and MTTI_T are retrieved from the MTTR database, if available, based on the entries for application description, unit type, LRU, and SRU.

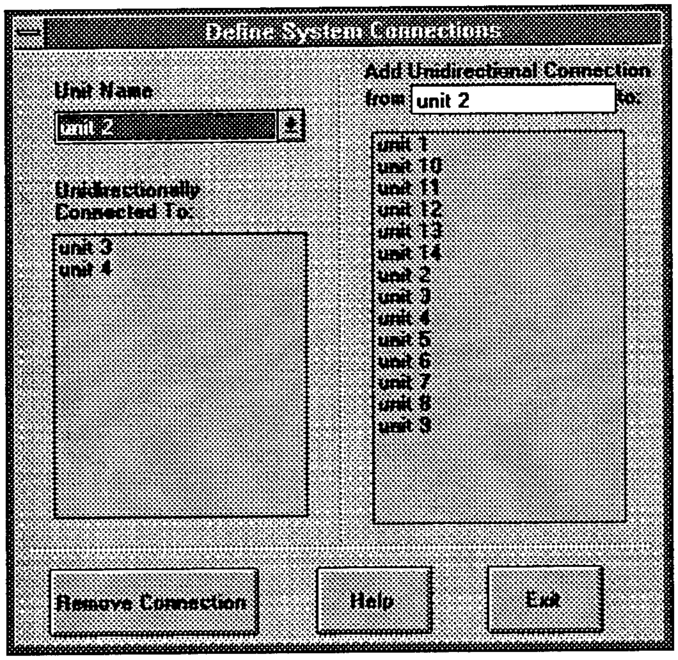
Performance requirement data and individual unit description data is entered through several radio buttons and check boxes. Criticality of performance, criticality of fault detection, technology, and circuit type should be selected through the appropriate radio buttons. Component units can be further described using the Component Description radio buttons. The Sequential checkbox should be used to indicate sequential or mostly sequential units. Commercial Off-the Shelf (COTS) units should be designated as such in the COTS checkbox. Each of these designations affects the DFT/BIT technique recommendations, as described in Section 4.2.3.

If the COTS checkbox is selected, an interface to the COTS reference database appears to allow the user to select the appropriate COTS part from the database. If the part does not exist in the database, the user is able to add the part to the database and immediately use it in the current project. (See Section 5.2.2.)

Mission requirement data is entered through five editboxes: MTTR, Reject Ratio, Failure Rate, Availability, and Expected Yield. These values will be used as input to the TFOM calculations, which will be performed each time MTTR, Failure rate, reject ratio, availability, or expected yield is changed, or when a new current unit is selected. The TFOMs will be displayed in the FFD, FFI, and E{AG} editboxes along the bottom of the dialog. The consistency of the FFD and FFI values will be checked when the dialog is exited.

5.2.1.2.2. System Connections Dialog

The second input dialog is the System Connections Dialog, shown in Figure 5.3, which is used to record the system interconnections. The connections are used in the DFT/BIT technique recommendations to find feedback loops and points of excessive fanout. All interconnections defined in TESPAD are considered unidirectional, i.e., the source and target are not interchangeable entities in the connections description.



Unit Name: unit 2

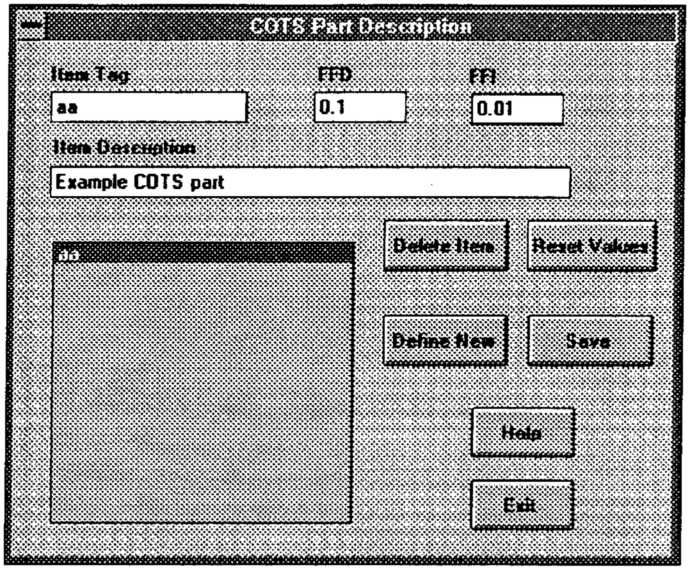
Unidirectionally Connected To:
unit 3
unit 4

Add Unidirectional Connection from unit 2 to:
unit 1
unit 10
unit 11
unit 12
unit 13
unit 14
unit 2
unit 3
unit 4
unit 5
unit 6
unit 7
unit 8
unit 9

Figure 5.3. System Connections Dialog

5.2.2. Reference Databases

Two reference databases assist the user by providing default data where applicable. A Commercial Off-the-Shelf (COTS) Parts database (see Figure 5.4) provides default data for Fraction of Faults Detected (FFD) and Fraction of Faults Isolated (FFI) for off-the-shelf components. A Mean Time to Repair (MTTR) database (see Figure 5.5) provides default data for MTTR and mean time to isolate values according to application description and unit type descriptions. An interface is provided for each of the reference databases to allow the user to view and edit the data in the databases.



Item Tag: aa
FFD: 0.1
FFI: 0.01
Item Description: Example COTS part

Figure 5.4. COTS Database Dialog

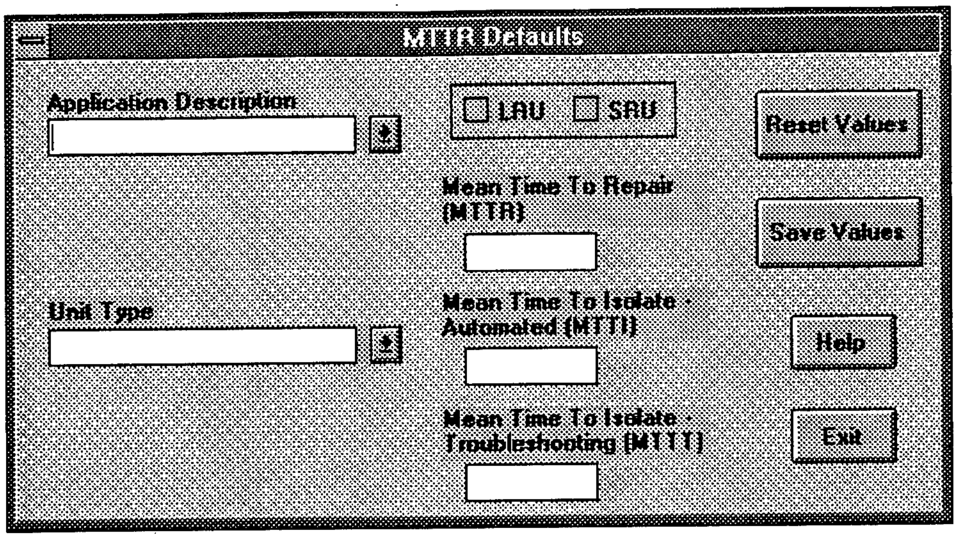


Figure 5.5. MTTR Database Dialog

5.2.3. Program Operation

TESPAD calculates Testability Figure of Merit (TFOM) values and produces DFT/BIT technique recommendations and metric and documentation templates based on system hierarchy, application data, mission and life-cycle requirement data, and reference databases according to the methodology developed in Task 4.

5.2.3.1. Testability Figure of Merit (TFOM) Calculations

TESPAD calculates values for the testability figures of merit, FFD, FFI, and E{AG}, based on the models described in Section 4.2.1. The calculations take place in the System Structure Dialog whenever one of the five inputs changes or a new current unit is selected. Results are calculated dynamically and are then displayed in the FFD, FFI, and E{AG} edit boxes of the System Structure Dialog. Required (non-zero) inputs for calculation of the metrics are: availability, failure rate, and MTTR (mean time to repair). Reject ratio and expected yield are also used in the calculation of the metrics, but these values are not required in order to perform the calculations. The default value for these two optional inputs is zero.

5.2.3.2. Consistency Check

As the System Structure Dialog is exited, a consistency check is performed on TFOMs for all units in the system. The check is designed to ensure that the top-down assignment of testability requirements is maintained. Starting at the lowest hierarchy level, a check value is calculated for FFD for each parent in the system as follows:

$$FFD_C = \sum_{i=1}^{n} FFD_i \left(\frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \right) \qquad (4\text{-}19)$$

where both summations (i and j) are performed over the number of child units, n, and λ_i is the failure rate of child i.

If the check value, FFD_c, exceeds the FFD value already calculated from the mission and life-cycle requirement data, the FFD value is replaced with the FFD_c value. If the FFD value exceeds the check value by more than 2%, an inconsistency warning is reported to the user, indicating that the value calculated for the parent cannot be attained using the children as specified. It is left to the user to remedy the problem. An identical check is performed for FFI values.

5.2.3.3. Technique Recommendations

TESPAD produces general and detailed Design-For-Test/Built-In Test technique recommendations for electronic systems. General recommendations are produced for the system and detailed recommendations are listed for each unit in the system based on hierarchy level, circuit type (analog/digital/mixed-signal), criticality of performance, criticality of fault detection, component description (if applicable), sequentiality, and the application environment.

The technique recommendations are produced by the tool in the context of *test architectures* to ensure a consistent test methodology throughout the system hierarchy. See Section 4.2.3. The test architectures are assigned to sub-hierarchies within the system according to the Test Architecture Assignment Algorithm, shown in Table 5.1, based on application environment (app), Criticality of Performance (CP), and Criticality of Fault Detection (CD).

Table 5.1. Test Architecture Assignment Algorithm

```
if ((CD==0 || app==6) && CP>0)                             TA = 1;
else if ((CD==0 && CP==0 && app>1) || (app==6 && CP==0))   TA = 2;
else if (CD==1 && (CP>0 || app==3 || app==5))              TA = 2;
else if (CD==0 && CP==0 && app<2)                          TA = 3;
else if (CD==1 && CP==0)                                   TA = 3;
else if (CD==2 && CP>0)                                    TA = 3;
else                                                       TA = 4;
```

Technique Recommendation Process

Test architecture assignment is only the first step in the technique recommendation process. After the test architectures have been assigned, the system topology is examined to determine intra-layer feedback loops and points of high fanout. This information allows the recommendation of techniques to be prioritized by unit so that the most critical testability implementation can be identified. DFT/BIT recommendations are prioritized through the process outlined in Section 4.2.3.

In the order given by the prioritizing step, techniques are assigned to units in the system by accessing technique scoreboards which were developed in Task 3. Each DFT/BIT technique has been evaluated over a number of different scoring criteria in order to establish an objective selection method in the recommendation process. Techniques were assigned a score of high, medium, or low on such criteria as: area overhead, power overhead, performance degradation, fault coverage, life-cycle usefulness, etc. Techniques were evaluated only on those criteria fault coverage, life-cycle usefulness, etc. Techniques were evaluated only on those criteria pertinent to their particular technique category. Each criterion was then scaled according to its effect at different life-cycle phases (R&D, manufacturing, and field) and also according to the target application environment (ground, air, or space). The resulting numerical scores were totalled for each application environment and stored, along with the individual criterion scoring, in the scoreboard databases. These scoreboard databases are used to field queries during the recommendation process to determine the most suited technique for the unit being analyzed according to the selection process outlined in Table 5.2.

All recommendations are printed to an output file in 80-column ASCII text, suitable for inclusion in a system specification. A general description of the overall test strategy suggested, including a listing of the test architectures defined in the system and a list of general design-for-test technique recommendations, precedes the detailed recommendations.

Table 5.2. Scoreboard Selection Table

Level of Assembly	Circuit Type	Component Description	Test Architecture	Selected Scoreboard
system	any			System-Level Testing
subsystem	any			System-Level Testing
board	digital		TA1	Digital Board-Level Testing and Fault Tolerance
board	digital		TA2,3,4	Digital Board-Level Testing
board	analog/mixed		TA1,2	Analog Board-Level BIT
board	analog/mixed		TA3	Analog Board-Level DFT
board	analog/mixed		TA4	Analog Board-Level Ad Hoc
component	digital	PLA	any	PLA BIT

Table 5.2. Scoreboard Selection Table

Level of Assembly	Circuit Type	Component Description	Test Architecture	Selected Scoreboard
component	digital	RAM	any	RAM BIT
component	digital	ROM	any	ROM BIT
component	digital	other	TA1	Structured Chip-Level BIT and Coding
component	digital	other	TA2	Structured Chip-Level BIT
component	digital	other	TA3	Structured Chip-Level DFT
component	digital	other	TA4	Ad Hoc DFT
component	analog/mixed	any	TA1,2	Analog Chip-Level BIT
component	analog/mixed	any	TA3	Analog Chip-Level DFT
component	analog/mixed	any	TA4	Analog Chip-Level Ad Hoc DFT

5.2.4. TESPAD Output

TESPAD produces testability requirements output in four basic forms: DFT/BIT technique recommendations, metric templates, documentation templates, and DFT/BIT technique templates. Each output is written to a text file which is available to the user for whatever manipulation is necessary to include in a SOW or system specification.

5.2.4.1. DFT/BIT Technique Recommendations

The DFT/BIT technique recommendations produced by TESPAD, as described above, contain general and detailed suggestions for DFT/BIT implementation within defined test architectures. The output produced by the recommendations process is a text file, in 80-column ASCII, containing an overview of the recommended testing strategy, a description of the test architectures used by TESPAD, a listing of the test architecture sub-hierarchies defined for the system, general design-for-test strategies, and specific testability technique recommendations for each unit in the system, ranked by the priority assigned in the recommendation process. A sample recommendation file is listed in Appendix IV.

5.2.4.2. Documentation Templates

Documentation templates suggest requirements for documentation of the testability implementation in the system at acceptance test. The suggestions include general requirements for each unit in the system on documentation of fault models and fault model assumptions used, documentation of fault sampling methods, documentation of operating environment requirements, documentation of test procedures, documentation of fault isolation strategies employed, and documentation of the test features included in any off-the-shelf components.

The documentation recommendations are produced in the order the units were created in the System Structure Dialog. Recommendations are based on the COTS status of each unit. For procured parts, documentation on the fault models, fault sampling, fault isolation strategy, test procedures and operation environment requirements is recommended. For COTS (Commercial Off-the-Shelf) parts, documentation on the operation environment requirements and test features included in the part is recommended. Also, for COTS parts, documentation on the fault models, fault sampling, fault isolation strategy, and test procedures is requested if available without additional cost. A sample documentation template file is listed in Appendix V.

5.2.4.3. Metric Templates

Metric templates, created at the user's request, describe the calculation used to arrive at each metric's value and suggested verification techniques for those metrics. The templates contain information such as recommended test phase, test means, and test mode, fault model assumptions, the quantitative definition of the metric, the quantitative requirement, the degree of allowable external support during testing, and the allowable requirements compliance tracking methodology. Descriptions for each calculated metric of each unit in the system are included in the file. A sample metric template is provided in Appendix VI.

5.2.4.4. DFT/BIT Technique Templates

DFT/BIT technique templates provide a "tutorial" description of each of the techniques referenced in the tool. The TESPAD techniques database includes a number of DFT/BIT techniques that can be applied to digital, analog, and mixed signal circuits. Overall, in excess of 145 techniques are included in the database.

The DFT/BIT technique templates include information on each technique covering such fields as: definition, description, generic design procedures for technique implementation, typical overhead and performance penalties, estimated coverage, compatibility issues, case histories, and references. Information on each of these techniques can be accessed through the DFT/BIT Techniques Templates output.

The DFT/BIT technique templates are produced through the DFT/BIT Technique Template Creation Dialog, shown in Figure 5.6. The output files, which are written in 80-column ASCII text, can be edited by the user in Notepad or can be imported into a word processor of the user's choice. A sample testability template is provided in Appendix VII.

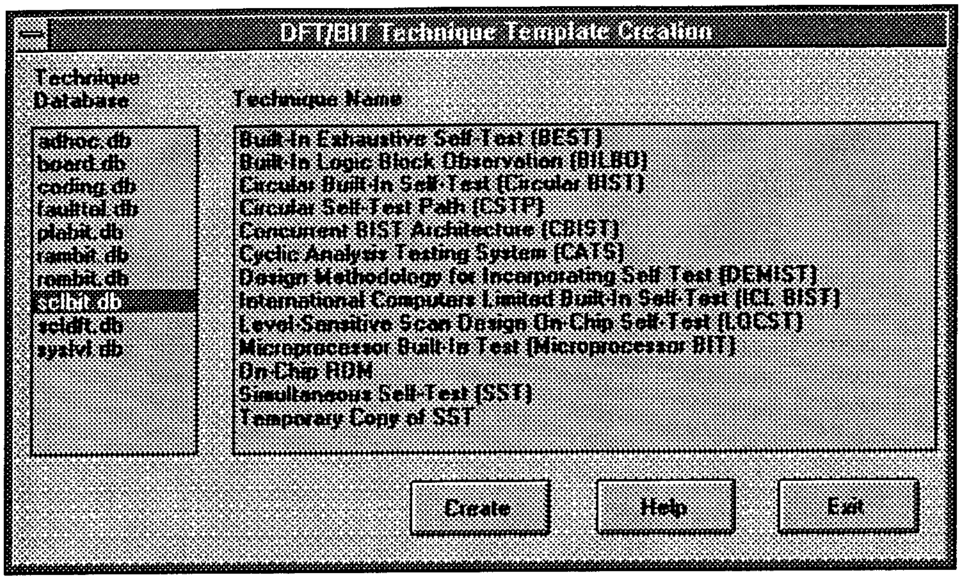


Figure 5.6. Technique Template Creation Dialog

5.3. TESPAD Development Environment

TESPAD was developed using Borland C++ 4.5, and the Borland Database Engine® 2.0. TESPAD contains approximately 7000 lines of C and C++ code utilizing Borland's Object Windows Library® for graphical interface development. All graphical interfaces were designed using the Borland Resource Workshop®.

All TESPAD databases (project databases, a COTS reference database, an MTTR reference database, DFT/BIT technique descriptive databases, DFT/BIT technique scoreboard databases) are maintained in the Borland Paradox® format. All database interactions are made possible through the use of the Borland Database Engine®.

TESPAD is a stand-alone product, requiring only Windows 3.1 or later; it does not require the use of any of the above Borland software packages.

5.4. TESPAD Operating Requirements

While the following operating requirements are not necessarily the minimum required to run TESPAD, it is suggested that these requirements be met:

A 486 personal computer
At least 4MB of RAM (preferably 8MB or more)
10MB of free hard disk space
A VGA monitor and VGA graphics card
A Windows-compatible mouse
Microsoft Windows 3.1 or later

5.5. TESPAD Testing

One of the most important phases in the development of software is test and validation. TESPAD was extensively tested throughout the development process by testing database interactions and functionality, interface functionality, TFOM calculation integrity, and overall program validity. Test cases were created and tested at several stages of the program.

TESPAD makes use of the Borland Database Engine® which is capable of interacting with several popular database formats including Paradox®, dBASE®, Quattro®. The Paradox® format is used exclusively in TESPAD. All database interactions were tested by executing typical database transactions on several sample database tables.

Each of the graphical interfaces in TESPAD was developed through the Borland Resource Workshop® and is accessed through the use of Borland's Object Windows Library®. Each interface was tested to verify its functionality and performance.

Several models are used in TESPAD to compute testability figures of merit (TFOMs) as outlined in Section 4.2.1. The implementations of each of these models was tested independently and collectively with the System Structure Dialog interface to determine proper functionality and to verify the results of the calculations.

Several test cases were created to verify the overall functionality as well as the specific requirements of TESPAD. Test cases were created both to approximate typical systems and to exercise specific TESPAD functions. Typical cases were developed with up to 40 units, using a large number of combinations of possible inputs. These typical cases were revised several times in attempts to crash the program. The causes of the few crashes that resulted were remedied. Several cases were developed to test specific portions of TESPAD operation. For example, one case, containing several convoluted feedback loops, was used to test the feedback loop identification function. Another case was developed to test the fanout check function. All functional operation was confirmed through the use of the test cases.

6. LESSONS LEARNED

As with any research endeavor, this project saw set-back and compromises. A recurring and dominant theme at each stage in the specification, design, and development of TESPAD was an inability to retrieve the types of empirical data necessary to make this a truly precise piece of software.

At the earliest stages, in developing a TFOM set that was realistic to use in specification and validation practices, it became clear that TFOM specifications are not well tracked during maintenance. Hence, the goal of determining realistically achievable levels of certain TFOMs was in part compromised, and could have been more severely so without the availability of industrial data to support our conclusions. This is especially true for analog and mixed-signal components and systems, for which empirical data is even more difficult to find than for digital logic.

This difficulty of obtaining empirical data was again echoed in the design of the specification translation models (from reliability system specifications to testability specifications). Models that we developed with the assumption of using empirical data from similar systems for certain inputs were thwarted by the unavailability of this information. For instance, one of the models for the calculation of FFI uses data on the mean time to isolate through automated means and the mean time to isolate though troubleshooting, as well as the mean time to replace a particular part. For at least certain components, we expected to be able to obtain this type of data, allowing us to use this, our most accurate, model for FFI. However, in the end, it was necessary to drop the mean time to replace component and drop the priority in using the model entirely, so that it is only called upon if the data is entered into a running database. Similar stories can be told for the other models used within TESPAD. In the end, while TESPAD employs accurate models with which we are satisfied, it does not have the highest level of precision for which we might have hoped. Rather, we have maintained the models we feel to be more precise and provided the database facilities necessary to, over time, build up the information set necessary to use them, so that some day TESPAD might operate at its optimum.

A recommendation we would make in regard to these difficulties is to implement better tracking of the types of data essential to specifying testability values. For instance, at a minimum, each system/assembly/board/component should be tracked at maintenance depots for its specified levels of TFOMs, the TFOM levels achieved in practice, whether or not it implements and DFT and/or BIT and what type (as a note, the maintenance personnel should also be made aware of this information and trained on how to use the DFT and BIT in a particular design, else it is wasted beyond production testing -- we found cases where this was evidently true), it's mean time to repair, mean time to isolate, mean time to troubleshoot if TPS or BIT methods fail, mean time to replace (for submodules in a system), production yield, acceptable reject ratio, failure rate/ mean time to failure, and cannot duplicate (CND) rate.

Certainly there are even more data fields that are relevant that are not even addressed here. As noted, the facilities to track some of this information have been implemented in TESPAD, and will need to be utilized to get the greatest benefit from the TESPAD software. Additionally, it is our understanding that at least some of this data has begun to be tracked by maintenance depots. We hope for the success of this endeavor so that future endeavors in this area will encounter fewer roadblocks.

7. CONCLUSIONS

The ultimate objective of this research project was the development of a structured testability implementation methodology that can be used by system developers to identify verifiable and cost-effective testability requirements in the procurement process of electronic systems and the incorporation of that methodology in a PC-based tool for use by those system developers. The implementation of that methodology was achieved in TESPAD, the Testability Specifications Advisor, a PC-based Windows 3.1 tool developed in this project and described in Section 5 of this report.

In the process of developing the methodology for testability implementation, it was first necessary to evaluate testability figures of merit (TFOMs) to assess which of the commonly used TFOMs are verifiable and cost-effective. Three of these commonly used TFOMs were identified as being both cost-effective and verifiable: Fraction of Faults Detected (FFD), Fraction of Faults Isolated (FFI), and Estimated Ambiguity Group Size (E{AG}). Translation models, used to specify these TFOMs based on reliability and performance requirements, were developed in the course of the evaluation process. The evaluation of these TFOMs is described in Section 2.

DFT and BIT techniques which can be used in a structured testability methodology to help achieve the TFOMs specified through the translation models above were identified and scored based on their cost-effectiveness. Over 150 digital and analog techniques were evaluated through this program. The evaluations are explained in Section 3 and detailed in the appendices.

Based on the TFOM and DFT/BIT technique evaluations above, a structured testability implementation methodology was developed which, given a set of functional system design requirements, the application environment and some information about the circuits involved in the different levels of the system hierarchy, defines and allocates detailed testability design and validation requirements including testability measures, recommended DFT/BIT methods and structures, data formats and data delivery, and validation/verification procedures. Through this methodology, described in Section 4, testability requirements are defined in a top-down process and DFT/BIT technique implementation, aimed at meeting those requirements, is allocated bottom-up. The end result is a testability design structure which specifies consistent, verifiable testability requirements and the testability means to achieve those requirements from the chip level to the system level of the design hierarchy.

The TESPAD tool is the manifestation of this structured testability methodology in a Windows-based, PC software tool. Through graphical interfaces, the user can input system design requirements, system hierarchy, and mission and performance related requirements, which are used by the tool to define the TFOM requirements and assign the DFT/BIT technique recommendations. TESPAD also develops validation and documentation requirements suitable for inclusion in system specifications. Reference databases, which accompany the tool, provide historical, default values that can, and should, be updated by the users of the tool. Tutorial information on each of the techniques referenced in the tool is provided and can be accessed at the user's discretion.

8. REFERENCES

[1] "Chip To System Testability", Research Triangle Institute, Interim Report to Rome Laboratory for Contract No. F30602-94-C0053. Available from Rome Laboratory, RL/ERSR, 525 Brooks Rd., Rome, NY 13441.

[2] B. Kaminska, Y. Savaria, "Design-for-Testability Using Test Design Yield and Decision Theory," Proceedings of the 1989 IEEE International Test Conference, 1989, pp. 884-892.

[3] S. Szygenda, "Profit, Liability, and Education: Influencing Factors on the Economics of Non-testing," Economics of Design and Test for Electronic Circuits and Systems, IEEE Press, 1992.

[4] Personal Communication from Frank Born, Rome Laboratory, September, 1994.

[5] C. Hawkins, et. al., "The VLSI Circuit Test Problem -- A Tutorial," IEEE Transactions on Industrial Engineering, v. 36, No. 2, Feb. 1989, pp. 111-116.

[6] D. Allen, "Failure Probability Algorithm for Test Systems to Reduce False Alarms," Proceedings of the 1990 International Test Conference, pp. 648-659.

[7] Personal Communications with Warner-Robbins Maintenance Technicians in the LNPA division, Oct. 24, 1994.

[8] D. Robb, "American Airlines Mobilizes Technicians," Avionics Magazine, April 1992, pp. 34-38.

[9] T. Moore, "A Test Process Optimization and Cost Modeling Tool," Proceedings of the 1994 International Test Conference, pp. 103-110.

[10] C. M. Maunder, *The Board Designer's Guide to Testable Logic Circuits*, Addison-Wesley Publishing Co., Workingham, England, 1992.

[11] H. Bleeker, P. van den Eijnden, and F. de Jong, *Boundary-Scan Test: A Practical Approach*, Kluwer Academic Pub., Dordrecht, The Netherlands, 1993.

[12] M. Davis, S. Kwan, and T. Holzer, "CADBIT II - Computer-Aided Design for Built-In Test," Hughes Missile Systems Company, Final Technical Report, June 1993.

[13] J. W. Watterson, M. Royals, and N. Kanopoulos, "Chip-Level Testability Requirements Guidelines," Research Triangle Institute, Final Technical Report, November 1992.

[14] B. W. Johnson, *Design and Analysis of Fault Tolerant Digital Systems*, Addison-Wesley Publishing Co., Reading, MA, 1989.

[15] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, New York, NY, 1990.

[16] R. G. Bennetts, *Design of Testable of Logic Circuits*, Addison-Wesley Publishing Co., London, England, 1984.

[17] T. R. N. Rao and E. Fujiwara, *Error-Control Coding for Computer Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[18] "Evaluation and Comparison of Techniques for Homogeneous Automatic Design for Testability," Research Triangle Institute, Final Report, March 1990.

[19] P. K. Lala, *Fault Tolerant & Fault Testable Hardware Design*, Prentice-Hall, Englewood Cliffs, NJ, 1985.

[20] K. A. Haller, K. Anderson, J. Zbytniewski and L. Bagnall, "Smart BIT," Grumman Aerospace Corporation, Final Technical Report, August 1985.

[21] C. Cooper, K. A. Haller, J. Zbytniewski, K. Anderson, and J. R. Matson, "Smart BIT-2," Grumman Aircraft Systems, Final Technical Report, December 1989.

[22] C. M. Maunder and R. E. Tulloss, *The Test Access Port and Boundary-Scan Architecture*, IEEE Computer Society Press, Los Alamitos, CA, 1990.

[23] P. Dennis, S. Vining, W. Daniel, and J. Marischen, "Test Bus Evaluation Report," Texas Instruments, Inc., January 1993.

[24] A. J. van de Goor, *Testing Semiconductor Memories: Theory and Practice*, John Wiley and Sons, Chichester, England, 1991.

[25] D. H. Merlino and J. Hadjilogiou, "Built-In Test Strategy for Next Generation Military Avionic Hardware," Proceedings of the International Test Conference, 1988, pp. 969-975.

[26] D. Das, S. Seth, and V. Agrawal, "Accurate Computation of Field Reject Ratio Based on Fault Latency," *Transactions on VLSI Systems*, v. 1, No. 4, Dec. 1993, pp. 537-545.

[27] D. Bellehsen, B. Kelley, A. Hanania, et. al., *Automated Testability Decision Tool,* Rome Laboratory Report RL-TR-91-200, Harris Corp., Sept. 1991.

[28] N. Fuqua, *Electronic Equipment Maintainability Data*, Reliability Analysis Center report EEMD-1, IIT Research Institute, Fall 1980.

[29] P. Bursch, et. al., "A PC-Based Expert Diagnostic Tool," AUTOTESTCON '87, pp. 73-77.

[30] R. Williams and C. Hawkins, "Errors in Testing," Proceedings of the 1990 International Test Conference, pp. 1018-1030.

Appendix I-A: Testability Measure Evaluation

Testability measures were identified for each category and evaluated for their applicability as system test specification metrics. The evaluation rated each measure based on five criteria:

1. Test Techniques to which the measure is applicable -- All, BIT only, scan only, DFT only, etc.
2. Usefulness -- Subjectively, how useful will the measure be in estimating the cost of testing a system?
3. Feasibility of making the measurement -- Can the measure be estimated/calculated in polynomial time, and if so, how feasible is it (subjectively) that the run-time will be acceptable?
4. Ambiguity -- Rated subjectively, can the measure be understood without qualifications?
5. Feasibility of specifying and verifying the quantity -- Rated subjectively, does it make sense to specify a quantity for the particular measure, and can the quantity be measured during acceptance testing?

Highly Recommended Testability Measures

Presence/absence of documentation of fault models

Presence/absence of documentation for fault sampling (if used)

Presence/absence of documentation of test vector generation methods

Presence/absence of documentation for operating environment requirements

Presence/absence of documentation of all test procedures used

Presence/absence of documentation of the fault isolation strategy employed

Presence/absence of documentation of the test features of any off-the-shelf components used

Percentage of BIT results that are statistically processed

Percentage of faults, for a given fault model, that are detected by BIT

Percentage of faults, for a given fault model, that are correctly isolated by BIT to an ambiguity group size of specified size or smaller

Presence/absence of a initialization methodology for test(s)

Percentage of board or higher-level nodes that can be probed

Presence/absence of hierarchical test structures

Percentage of components with a boundary scan chain and test access port

Percentage of circuit boards with a test bus

Presence/absence of a subsystem bus that is accessible during test

Presence/absence of a system bus that is accessible during test

Percentage of component I/O with boundary scan cells

Percentage of boundary scan components that support a BYPASS instruction

Presence/absence of a hierarchical fault isolation methodology

Percentage of commercial components used that have Boundary Scan and/or DFT or BIT features

Percentage of sequential logic that is scannable

Percentage of bus signals that may be tri-stated

Percentage of sequential logic that is asynchronous

Percentage of clock signals that can be independently controlled during test

Percentage of nodes accessible to a connector or lead

Percentage of redundant logic that can be independently tested

Percentage of detected faults, for a given fault model, that are correctly isolated to an ambiguity group size of specified size or smaller

Percentage of faults detected for a given fault model

Average ambiguity group size

Appendix I-B: DFT/BIT Evaluations

I-B.1. Technique Identification

Fourteen categories of testability techniques were defined, covering both a variety of chip-level structures and spanning the hierarchy of implementation levels. These categories considered were:

A. Structured Chip-Level BIT
B. Structured Chip-Level DFT
C. Ad-hoc DFT
D. RAM BIT
E. ROM BIT
F. PLA BIT
G. Coding
H. Fault Tolerance
I. Board-Level Testing
J. System-Level Testing
K. Intelligent BIT
L. Analog BIT
M. Analog Structured DFT
N. Analog Ad-hoc DFT

Within these categories, 133 of the candidate DFT and BIT techniques were found to be feasible for real-world implementations, in this case, indicating that their overhead required or probable performance was not so obviously limiting as to discount them from consideration. The techniques considered were:

A1. Concurrent BIST Architecture (CBIST)
A2. Cyclic Analysis Testing System (CATS)

A3. Circular BIST
A4. Circular Self-Test Path (CSTP)
A5. Built-In Logic Block Observation (BILBO)
A6. Built-In Exhaustive Self-Test (BEST)
A7. Design Methodology for Incorporating Self-Test (DEMIST)
A8. LSSD On-Chip Self-Test (LOCST)
A9. Simultaneous Self-Test (SST)
A10. International Computers Limited BIST (ICL BIST)
A11. On-Chip ROM
A12. Microprocessor BIT

B1. Level-Sensitive Scan Design (LSSD)
B2. Weighted Random Test
B3. Pseudo-Exhaustive Test
B4. CrossCheck
B5. Quiescent Power Supply Current Test (IDDQ)

C1. Provide initialization capability
C2. Provide global feedback path control
C3. Provide partitioning of large circuits
C4. Add test points to enhance controllability and observability
C5. Test embedded structures separately
C6. Separate mixed logic families
C7. Limit chip fanout at test points to one less than maximum
C8. Disable free running system clocks during test
C9. Disable internal one-shots during test
C10. Avoid redundant logic

C11. Avoid wired AND/OR functions
C12. Avoid asynchronous design practices
C13. Avoid clock gating

D1. BIST for Embedded Static RAMs
D2. Modified Algorithm Test Sequence BIT
D3. Parallel Test
D4. BIST Implementation of Parallel Test Algorithm
D5. Exhaustive Random Sequence (bidirectional)
D6. Exhaustive Random Sequence (unidirectional)
D7. Exhaustive Random Sequence (random checker)
D8. Walking 1/0 Test
D9. MATS++
D10. March C-
D11. Jacobson's Test

E1. Count-Based Compaction
E2. Parity-Based Compaction
E3. Polynomial-Division-Based Compaction
E4. Exhaustive Enhanced Output Data Modification (EEODM)

F1. Concurrent Test Using Error Detection Code
F2. Design with Cumulative Parity Comparison
F3. Divide and Conquer Strategy
F4. Test with Low Overhead and High Fault Coverage
F5. Exhaustive Test with Signature Analysis

F6. BIST Input Generator
F7. Self-Checking Design Using Alternating Logic

G1. Cyclic Codes
G2. Hamming Codes
G3. Arithmetic Codes
G4. Checksums
G5. Berger Codes
G6. Parity Codes
G7. Duplication Codes

H1. Duplication with Comparison
H2. Static Redundancy
H3. Dynamic Redundancy (Operative Standby)
H4. Dynamic Redundancy (Inoperative Standby)
H5. Hybrid Redundancy
H6. Time Redundancy
H7. Differential Cascode Voltage Switch Design (DCVS)

I1. Functional Test
I2. On-Board Pseudo-Random Pattern Generator and Multiple-Input Shift Register
I3. On-Board Integration of VLSI Chip BIT
I4. On-Board ROM
I5. In-Circuit Test
I6. IEEE Standard 1149.1

J1. IEEE Proposed Standard 1149.5
J2. IEEE Standard 488.1

K1. Integrated BIT
K2. Adaptive BIT
K3. Temporal Monitoring BIT

L1. For input stimulus generation, use constant (DC) voltage sources
L2. For input stimulus generation, use constant current sources
L3. For input stimulus generation, use sine wave generators
L4. For input stimulus generation, use square wave generators
L5. For input stimulus generation, use triangle wave generators
L6. For input stimulus generation, use sawtooth wave generators
L7. For input stimulus generation, use monolithic waveform generators
L8. For input stimulus generation, use noise diode sources
L9. For input stimulus generation, use a pseudorandom pattern gen. and D/A conversion
L10. For input stimulus generation, use a digitized waveform and D/A conversion
L11. For coupling a stimulus source to a circuit, use existing antennas
L12. For coupling a stimulus source to a circuit, use analog multiplexing
L13. For coupling a stimulus source to a circuit, use RF relays
L14. For coupling a stimulus source to a circuit, use directional couplers
L15. For coupling a stimulus source to a circuit, use transformers
L16. For output response evaluation, use level detectors
L17. For output response evaluation, use frequency detectors
L18. For output response evaluation, use comparators
L19. For output response evaluation, use window detectors
L20. For output response evaluation, use peak detectors

L21. For output response evaluation, use phase detectors
L22. For output response evaluation, use sample and hold circuits
L23. For output response evaluation, use A/D conversion and a processor
L24. For output response evaluation, use A/D conversion and a magnitude accumulator
L25. For coupling a circuit to a response evaluator, use existing antennas
L26. For coupling a circuit to a response evaluator, use analog multiplexing
L27. For coupling a circuit to a response evaluator, use RF relays
L28. For coupling a circuit to a response evaluator, use directional couplers
L29. For coupling a circuit to a response evaluator, use transformers

M1. For mixed signal chips, use internal scan path in the digital section
M2. For mixed signal circuit boards, use boundary scan (IEEE 1149.1) in the digital section
M3. For mixed signal circuits, use analog multiplexing techniques for partitioning
M4. For analog circuits, use analog "scan path" techniques
M5. For analog circuit boards, use analog test buses (IEEE P1149.4)

N1. Use physical partitioning
N2. Use electronic partitioning
N3. Use test points for controllability, observability, and calibration
N4. Use buffering techniques for test points
N5. Ensure proper size, shape, and location of test pads
N6. Use adequate ground pads
N7. Ensure proper and compatible test connector design
N8. Ensure no high voltage or high frequency signals exist in the test interface
N9. Ensure direct test access to A/D and D/A converters
N10. Ensure direct test access to transducers
N11. Ensure direct test access to driver elements

N12. Break global feedback paths
N13. Break long functional paths
N14. Isolate complex circuits or components for separate test
N15. Inhibit free-running clocks
N16. Avoid single points of failure
N17. Minimize manual adjustments
N18. Terminate unused inputs
N19. Avoid select-on-test components
N20. Avoid electromechanical devices
N21. Ensure UUT/ATE interface integrity and compatibility
N22. Use worst case design practices
N23. Ensure reasonable and accurate test specifications

I-B.2 Evaluation Criteria

Evaluation criteria were next developed to assess the specific benefits and trade-offs for each particular technique. Fourteen fundamental evaluation criteria were developed, focusing on the areas of technique design, capability and performance, and application. These are listed and defined below.

Area Overhead

At points in the hierarchy above chip-level, a modified version of this criterion is used which represents the total hardware required for the use of the technique. Lower overhead results in a higher score.

I/O Overhead

Lower overhead results in a higher score.

Power Overhead

Lower overhead results in a higher score.

Computational Expense of Design Translation

Lower computational expense results in a higher score.

Performance Degradation

Lower degradation results in a higher score.

Test Run Time

Lower test run time results in a higher score.

Stuck-at Fault Coverage

A number of technique categories use variations of this criterion, as some classes of DFT/BIT are designed with other fault models in mind. Any modifications to the above definition are explained when introduced. Higher coverage results in a higher score.

Delay, Open, and Bridging Fault Coverage

Some categories use variations of this criterion to describe technique behavior in the presence of other types of faults relevant to their environment. Any modifications to the above definition are explained when introduced. Higher applicability or coverage results in a higher score.

Impact on Reliability, Availability, and Maintainability

A more positive impact results in a higher score.

Applicability to Various Circuit Structures

Higher applicability results in a higher score.

Compatibility with Other DFT/BIT Structures

Higher compatibility results in a higher score.

Compatibility with IEEE Standard 1149.1 and Other Test Buses

Higher compatibility results in a higher score.

ATE/Diagnostics Accessibility

Higher accessibility results in a higher score.

Life-Cycle Usefulness

Techniques which can be used at many levels of testing throughout the product life-cycle merit higher scores.

I-B.3 Candidate Technique Evaluations

Finally, candidate DFT/BIT techniques were evaluated with respect to the evaluation criteria. For each of the fourteen technique categories, a set of evaluation criteria was created by selecting from the original set, with some slight alterations as necessary, to reflect the relevant features taken into consideration when selecting a test technique from the given category. In the case that a criterion did not apply, or when all such techniques had very similar features with respect to a criterion, it was dropped from consideration in that category.

Information about each technique, including a general description, case histories where available, literature references, and information regarding the applicable criteria, was entered into a database, for accessibility by the software tool. This information was used to assign each technique a score for each criterion. The scores used were Low, Medium, and High (L, M, and H), with pluses and minuses added where more detailed differentiation could be justified. In a few cases, a score of "v" (variable) was assigned, indicating a rating which is highly dependent on the implementation. As a reminder, those criteria for which low values result in high scores have bars over their numbers in the tables (meaning that a score of 'L' is more favorable than a score of 'H'), e.g., area overhead. The technique evaluation tables, including a listing of the applicable evaluation criteria for each technique category, are presented below.

A. Structured Chip-Level BIT

1. Area Overhead
2. I/O Overhead
3. Power Overhead
4. Computational Expense of Design Translation
5. Performance Degradation
6. Test Run Time
7. Stuck-at Fault Coverage
8. Delay/Open/Bridging Fault Coverage
9. Impact on Availability, Reliability, Maintainability
10. Applicability to Various Circuit Structures
11. Compatibility with Other DFT/BIT Structures
12. Compatibility with IEEE Std 1149.1 and Other Test Buses
13. ATE/Diagnostics Accessibility
14. Life-Cycle Usefulness

Table I-1. Structured Chip-level BIT Evaluation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Concurrent BIST Architecture (CBIST)	L	M	L	M	M	M	H	M	M	M	M	M	M	M
Cyclic Analysis Testing System (CATS)	H	M	H	H	H	M	H	M	M	M	M	M	M	M
Circular BIST	M	M	M	L	M	M	H	M	M	M	M	M	H	M
Circular Self-Test Path (CSTP)	M	M	M	L	M	M	H	M	M	M	M	M	M	L
Built-In Logic Block Observation (BILBO)	L	M	L	M	L	M	H	M	M	M	M	M	H	M
Built-In Exhaustive Self-Test (BEST)	L	M	L	L	M	M	H	M	M	L	M	M	H	M
DEMIST	L	M	L	L	L	M	H	M	M	H	M	M	H	M
LSSD On-Chip Self-Test (LOCST)	M	M	M	M	M	L	H	M	M	M	M	M	H	H
Simultaneous Self-Test (SST)	M	M	M	M	M	H	H	M	M	M	M	M	M	M
International Computers Limited BIST	L	L	L	L	H	M	H	M	M	H	M	L	H	M
On-Chip ROM	L	H	L	M	H	H	H	H	M	M	M	M	M	M
Microprocessor BIT	v	H	v	M	H	M	H	H	v	L	M	M	H	H

B. <u>Structured Chip-Level DFT</u>

1. Area Overhead
2. I/O Overhead
3. Power Overhead
4. Computational Expense of Design Translation
5. Performance Degradation
6. Test Run Time
7. Stuck-at Fault Coverage
8. Delay/Open/Bridging Fault Coverage
9. Impact on Availability, Reliability, Maintainability
10. Applicability to Various Circuit Structures
11. Compatibility with Other DFT/BIT Structures
12. Compatibility with IEEE Std 1149.1 and Other Test Buses
13. ATE/Diagnostics Accessibility
14. Life-Cycle Usefulness

Table I-2. Structured Chip-level DFT Evaluation

	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	7	8	9	10	11	12	13	14
Level-Sensitive Scan Design (LSSD)	M	M	M	M	H	M	H	M	H	M	H	M	H	M
Weighted Random Test	M	M	M	M	H	M	H	M	M	M	H	M	M	M
Pseudo-Exhaustive Test	M	M	H	M	H	M	H	M	M	H	M	M	M	H
CrossCheck	M	M	H	L	H	M	H	M	H	M	H	H	H	H
Quiescent Power Supply Current Test (IDDQ)	M	M	M	M	M	H	H	M	M	L	H	M	M	M

C. Ad-hoc DFT

1. Area Overhead
2. I/O Overhead
4. Computational Expense of Design Translation
5. Performance Degradation

7d. Impact on Fault Coverage/Test Pattern Generation Time - the amount of fault coverage that can be achieved for a given test set length.

Table I-3. Digital Ad Hoc DFT Evaluation

	$\overline{1}$	$\overline{2}$	$\overline{4}$	$\overline{5}$	7d
Provide initialization capability	H	H	H	H	H
Provide global feedback path control	H	H	H	M	M
Provide partitioning of large circuits	M	M	M	M	H
Add test points to enhance controllability and observability	M	L	M	M	H
Test embedded structures separately	L	M	L	M	H
Separate mixed logic families	M	H	M	M	H
Limit chip fanout at test points to one less than maximum	H	H	H	H	M
Disable free running system clocks during test	H	H	H	M	H
Disable internal one-shots during test	H	H	H	M	M
Avoid redundant logic	H	H	H	M	H
Avoid wired AND/OR functions	M	H	M	H	H
Avoid asynchronous design practices	H	H	M	M	H
Avoid clock gating	M	H	L	H	H

D. RAM BIT

1. Area Overhead
4. Computational Expense of Design Translation
5. Performance Degradation
6. Test Run Time
7a. Cell Stuck-at Fault Coverage - fault model: bit storage cell stuck independently at 1 or 0.
8a. Cell Coupling Fault Coverage - fault model: value stored in cell depends on other cell states; pattern sensitive faults.
8b. Decoder Fault Coverage - fault model: memory access controls incorrect or multiple locations.
14. Life-Cycle Usefulness

Table I-4. RAM BIT Evaluation

	$\overline{1}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	7a	8a	8b	14
BIST for Embedded Static RAMs	L	M+	M	M	H	H	H	M
Modified Algorithm Test Sequence BIT	M−	M	M	M+	H	L	H	H
Parallel Test	M	M−	H−	H	H	L	H	M
BIST Implementation of Parallel Test Algorithm	M	M−	H−	H	H	M	H	M
Exhaustive Random Sequence (bidirectional)	M	H−	H	M+	H	L	H	M+
Exhaustive Random Sequence (unidirectional)	H−	H	H	M+	H	L	H−	M+
Exhaustive Random Sequence (random checker)	H	H	H	M	H	L	H−	M+
Walking 1/0 Test	H	H	H	L	H	H	H	L
MATS++	H	H	H	M+	H	L	H	M
March C-	H	H	H	M	H	M	H	M
Jacobson's Test	H	H	H	M+	H	L	H	M

<u>E. ROM BIT</u>

1. Area Overhead
4. Computational Expense of Design Translation
6. Test Run Time
7b. Fault Coverage - fault model: anything which yields incorrect data at ROM outputs.

Table I-5. ROM BIT Evaluation

	$\overline{1}$	$\overline{4}$	$\overline{6}$	7b
Count-Based Compaction	M	M+	H	M
Parity-Based Compaction	H−	M+	H	M−
Polynomial Division-Based Compaction	H−	M+	H	M
Exhaustive Enhanced Output Data Modification (EEODM)	M	M	H−	H

F. PLA BIT

1. Area Overhead
2. I/O Overhead
4. Computational Expense of Design Translation
5. Performance Degradation
6. Test Run Time

7c. Single Stuck-at Fault Coverage - fault model: single line stuck-at 1 or 0.

8c. Single Crosspoint Fault Coverage - fault model: addition or subtraction of a single crosspoint connection.

8d. Single Bridging Fault Coverage - fault model: two signal lines connecting together inappropriately.

8e. Multiple Fault Coverage - fault model: any combination of two or more of the above faults.

9. Impact on Availability, Reliability, Maintainability

Table I-6. PLA BIT Evaluation

	$\overline{1}$	$\overline{2}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	7c	8c	8d	8e	9
Concurrent Test Using Error Detection Code	M−	M+	L	H	H	H	H	?	?	H
Design with Cumulative Parity Comparison	M	M−	M	M	H	H	H	H	H	M−
Divide and Conquer Strategy	M	M−	M	M	M	H	H	?	H	M−
Test with Low Overhead and High Fault Coverage	M+	M	M	M	H	H	H	?	H	M−
Exhaustive Test with Signature Analysis	M+	M+	H	H	L	H	H	H	H	M
BIST Input Generator	M	M+	H−	M	M	H	H	H	?	M
Self-Checking Design Using Alternating Logic	M	M	H	M	H	H	H	H	?	H

G. Coding

1. Area Overhead
4. Computational Expense of Design Translation
5. Performance Degradation
7e. Detection Capability - the ability of the code to detect bit errors.
8f. Correction Capability - the ability of the code to correct bit errors.
10. Applicability to Various Circuit Structures

Table I-7. Coding Evaluation

	$\overline{1}$	$\overline{4}$	$\overline{5}$	7e	8f	10
Cyclic Codes	M	M	M	H	H	L
Hamming Codes	M	M	M	M	H	M
Arithmetic Codes	M	M	M	H	M	L
Checksums	M	M	H	H	L	L
Berger Codes	M	M	M	H	L	M
Parity Codes	H	H	H	L	L	M
Duplication Codes	L	M	M	H	L	H

<u>H. Fault Tolerance</u>

1. Area Overhead
4. Computational Expense of Design Translation
5. Performance Degradation
9. Impact on Availability, Reliability, Maintainability
10. Applicability to Various Circuit Structures

Table I-8. Fault Tolerance Evaluation

	$\overline{1}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	9	10
Duplication with Comparison	M	M	H	H−	M	H
Static Redundancy	M	L	M	H	H	H
Dynamic Redundancy (Operative Standby)	L+	L	M	H−	H−	H
Dynamic Redundancy (Inoperative Standby)	L+	H	L	M	M	H
Hybrid Redundancy	L	L	L	H	H	H
Time Redundancy	H	H	M	L	M	H
Differential Cascode Voltage Switch Design (DCVS)	M−	M	M	H	H	L

I. Board-Level Testing

1a. Overhead - the total hardware required for the use of the technique.
4. Computational Expense of Design Translation
5. Performance Degradation
6. Test Run Time
7f. Fault Detection - the ability of the technique to detect faults of any type.
9. Impact on Availability, Reliability, Maintainability
11a. Compatibility with Chip-Level DFT/BIT Structures - the compatibility of the technique with DFT/BIT structures at lower levels of hierarchy.
13. ATE/Diagnostics Accessibility
14. Life-Cycle Usefulness

Table I-9. Board-level Testing Evaluation

	$\overline{1a}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	7f	9	11a	13	14
Functional Test	H	H	H	M	L	M	M	L	L
On-Board PRPG/MISR	L+	H−	M	L	H−	M	M	M	H
On-Board Integration of VLSI Chip BIT	L	L	H	L	H	M	H−	M	H
On-Board ROM	L	M	M	H−	H	M	M	M+	H
In-Circuit Test	H	H	H	H	M	M	M	M+	M
IEEE Standard 1149.1	M	M	M	M	H	H	H	H	H

J. System-Level Testing

1a. Overhead - the total hardware required for the use of the technique.
4. Computational Expense of Design Translation
5. Performance Degradation
6. Test Run Time
7f. Fault Detection - the ability of the technique to detect faults of any type.
9. Impact on Availability, Reliability, Maintainability
11b. Compatibility with Board-Level DFT/BIT Structures - the compatibility of the technique with DFT/BIT structures at lower levels of hierarchy.
13. ATE/Diagnostics Accessibility
14. Life-Cycle Usefulness

Table I-10. System-level Testing Evaluation

	— 1a	— 4	— 5	— 6	7f	9	1 1b	1 3	1 4
IEEE Proposed Standard 1149.5	M	M	H	H	H	H	H	H	H
IEEE Standard 488.1	M	H	H	H	M	M	H	H	H

K. Intelligent BIT

1a. Overhead - the total hardware required for the use of the technique.
4. Computational Expense of Design Translation
5. Performance Degradation
9. Impact on Availability, Reliability, Maintainability
14. Life-Cycle Usefulness

Table I-11. Intelligent BIT Evaluation

	— 1a	— 4	— 5	9	1 4
Integrated BIT	M	M	M	H-	H
Adaptive BIT	M	M+	H	H	M
Temporal Monitoring BIT	H	H	M	M	H

L. Analog BIT

1. Area Overhead
2. I/O Overhead
3. Power Overhead
4. Computational Expense of Design Translation
5. Performance Degradation
6. Test Run Time
7. Stuck-at Fault Coverage
8. Delay/Open/Bridging Fault Coverage
9. Impact on Availability, Reliability, Maintainability
10. Applicability to Various Circuit Structures
11. Compatibility with Other DFT/BIT Structures
12. Compatibility with IEEE Std 1149.1 and Other Test Buses
13. ATE/Diagnostics Accessibility

14. Life Cycle Usefulness

Table I-12. Analog BIT Evaluation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Input: use constant (DC) voltage sources	L	M	M	L	M	H	M	M	M	M	M	H	H	H
Input: use constant current sources	L	M	M	L	M	H	M	M	M	M	M	H	H	H
Input: use sine wave generators	L	M	M	L	M	H	M	M	M	L	L	H	H	H
Input: use square wave generators	L	M	M	L	M	H	M	M	M	L	L	H	H	H
Input: use triangular wave generators	L	M	M	L	M	H	M	M	M	L	L	H	H	H
Input: use sawtooth wave generators	L	M	M	L	M	H	M	M	M	L	L	H	H	H
Input: use monolithic waveform generators	L	M	M	L	M	H	M	M	M	L	L	M	H	H
Input: use noise diode sources	M	M	H	L	L	H	L	L	M	L	L	L	M	M
Input: use a PRPG and D/A	L	M	M	L	M	H	L	L	L	L	L	L	M	M
Input: use a digitized waveform and D/A	L	M	M	L	M	H	L	L	L	L	L	M	M	M
Input coupling: use existing antennas	M	H	H	H	H	H	L	L	M	L	L	L	M	M
Input coupling: use analog multiplexing	L	L	M	L	H	M	L	L	M	L	L	M	H	H
Input coupling: use RF relays	L	M	L	L	L	M	L	L	L	L	L	L	M	M
Input coupling: use directional couplers	M	M	L	M	L	M	L	L	L	L	L	L	M	M
Input coupling: use transformers	L	M	M	L	L	M	L	L	M	L	L	L	H	M
Output: use level detectors	L	M	M	L	M	H	M	M	M	M	L	H	H	H
Output: use frequency detectors	L	M	M	L	M	H	M	M	M	L	L	H	H	H
Output: use comparators	L	M	M	L	M	H	M	M	M	M	L	H	H	H
Output: use window detectors	L	M	M	L	M	H	M	M	M	M	L	H	H	H
Output: use peak detectors	L	M	M	L	M	H	M	M	M	M	L	H	H	H
Output: use phase detectors	L	M	M	L	M	H	M	M	M	L	L	H	H	H
Output: use sample and hold circuits	L	M	M	L	M	M	M	M	M	M	L	M	H	H
Output: A/D and use processor	L	L	M	L	M	H	L	L	L	M	L	H	H	H

Output: A/D and use magnitude accumulator	L	L	M	L	M	H	L	L	L	M	M	H	H	H
Output coupling: use existing antennas	M	H	H	M	H	M	L	L	M	L	L	L	M	M
Output coupling: use analog multiplexing	L	M	M	L	M	M	L	L	M	L	L	H	H	H
Output coupling: use RF relays	L	M	L	L	M	M	L	L	L	L	L	L	M	M
Output coupling: use directional couplers	L	M	M	L	M	M	L	L	M	L	L	L	M	M
Output coupling: use transformers	L	M	M	L	L	M	L	L	M	L	L	L	H	H

M. Analog Structured DFT

1. Area Overhead
2. I/O Overhead
3. Power Overhead
4. Computational Expense of Design Translation
5. Performance Degradation
6. Test Run Time
7. Stuck-at Fault Coverage
8. Delay/Open/Bridging Fault Coverage
9. Impact on Availability, Reliability, Maintainability
10. Applicability to Various Circuit Structures
11. Compatibility with Other DFT/BIT Structures
12. Compatibility with IEEE Std 1149.1 and Other Test Buses
13. ATE/Diagnostics Accessibility
14. Life Cycle Usefulness

Table I-13. Analog Structured DFT Evaluation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Mixed signal chips: use internal scan path	L	M	M	M	M	M	M	M	M	L	M	M	L	L
Mixed signal circuit boards: use boundary-scan	L	M	M	M	M	H	H	H	M	L	M	M	L	M
Mixed signal circuits: use analog multiplexing	L	M	M	L	M	H	M	M	L	L	L	L	L	M
Analog circuits: use analog "scan path"	L	M	M	L	M	M	M	M	M	L	L	L	L	M
Analog circuit boards: use analog test buses	L	M	M	L	M	H	H	H	M	L	M	M	L	M

N. Analog Ad-hoc DFT

1. Area Overhead
2. I/O Overhead
3. Power Overhead
4. Computational Expense of Design Translation
5. Performance Degradation
6. Test Run Time
7. Stuck-at Fault Coverage
8. Delay/Open/Bridging Fault Coverage
9. Impact on Availability, Reliability, Maintainability
10. Applicability to Various Circuit Structures
11. Compatibility with Other DFT/BIT Structures
12. Compatibility with IEEE Std 1149.1 and Other Test Buses
13. ATE/Diagnostics Accessibility
14. Life Cycle Usefulness

Table I-14. Analog Ad Hoc DFT Evaluation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Use physical partitioning	L	M	L	M	M	H	H	M	M	H	H	H	M	M
Use electronic partitioning	L	M	L	M	M	H	H	M	M	H	H	H	M	M
Use test points for control/observe/calibration	M	L	M	L	H	L	M	H	M	H	H	H	M	H
Use buffering techniques for test points	L	M	L	M	H	M	L	L	M	M	H	H	L	H
Ensure proper size, shape, location of test pads	M	M	M	M	H	H	M	M	M	M	H	H	L	M
Use adequate ground pads	M	M	M	M	H	H	L	L	M	M	H	H	L	M
Ensure proper test connector design	L	H	H	M	H	M	H	M	M	M	H	M	M	H
Ensure no high V/f signals in the test interface	H	H	H	M	H	M	M	L	H	H	M	M	L	H
Ensure direct test access to converters	M	M	H	L	M	M	H	H	L	M	H	H	M	H
Ensure direct test access to transducers	M	L	M	L	M	M	H	H	M	M	H	H	M	H
Ensure direct test access to driver elements	M	L	M	L	M	L	H	H	M	M	H	H	M	M
Break global feedback paths	L	L	L	L	M	L	M	M	M	M	H	H	M	M
Break long functional paths	L	L	L	L	M	L	M	M	M	M	H	H	M	M
Isolate complex circuits for separate test	L	L	L	L	L	L	H	H	H	H	M	H	H	M

Inhibit free-running clocks	M	M	M	H	L	M	H	M	M	M	L	H	M	M
Avoid single points of failure	L	H	L	L	L	L	L	M	L	L	L	M	L	M
Minimize manual adjustments	H	H	M	M	H	H	M	M	M	L	H	M	L	L
Terminate unused inputs	M	H	M	H	H	H	M	M	M	M	H	M	M	L
Avoid select-on-test components	H	M	M	M	H	H	L	H	M	L	H	M	H	L
Avoid electromechanical devices	H	M	H	M	H	H	L	M	H	L	H	M	L	M
Ensure UUT/ATE interface integrity	M	M	M	M	M	H	M	M	M	M	H	H	H	H
Use worst case design practices	M	M	L	L	M	H	L	L	M	M	M	L	L	L
Ensure reasonable/accurate test specifications	H	L	M	M	M	H	M	M	M	M	H	M	M	H

Appendix II: Practical Levels of TFOMs

II-1 Practical Levels of Digital Circuit TFOMs

II-1.1 FFD

Empirical data collected for Tasks 1 and 2 were combined to yield an estimation of practical TFOM levels for various levels of assembly and various levels of complexity. The data pertaining to FFD is summarized in Table II-1.

Table II-1. Achieved Levels of FFD for Digital Logic

Level of Assembly	**Complexity (L, M, H)**	**Achieved FFD (%)**	**Cost (L, M, H)**
Component	L	97.5	M+
	L	90	L
	M	92.5	L/M
	M	95	M+
	M	95	M+
	M	97.5	H
	M	77.5	L
	M	60	L
	M	84-89	L
	M	84-90	L
	M	93.3	H
	H	98-100	H
	H	97.5	M
	H	97.5	M+
	H	85	M+

Table II-1. Achieved Levels of FFD for Digital Logic (Continued)

Level of Assembly	**Complexity (L, M, H)**	**Achieved FFD (%)**	**Cost (L, M, H)**
Component	H	96	M+
	H	60	M
	H	90	M
Component/ Board	N/A	60	L
	N/A	60-85	M
	N/A	85+	H
Board	M	70-99	M/H
	M+/H	99.65	H
	M+/H	99.73	H
	M+/H	99.9	H
	M+/H	99.9	H
	H	95	M+
System	N/A	70	L
	N/A	70-90	M
	N/A	90+	H
	N/A	95	H

From this data, we established low, medium, and high levels for FFD, relative to the cost of attaining those levels, the level of assembly, and (for components) the complexity of the design. Based on the data collected, the FFD relations shown in Table II-2 were developed.

Table II-2. Practical Levels of FFD for Digital Logic

Level of Assembly	Complexity (L, M, H)	Low Cost FFD (%)	Medium Cost FFD (%)	High Cost FFD (%)
Component	L	60-90	90-97	98+
	M	60-90	90-95	95+
	H	50-80	80-95	95+
Board	N/A	<70	70-90	90-95+
System	N/A	<70	70-85	85-90+

The cost data was broken out further into costs incurred for components and systems implementing DFT, components and systems implementing BIT (possibly including DFT), and components and systems tested by conventional means. Divided along these lines, we obtained the breakdown shown in Tables II-3 through II-5.

Table II-3. Achieved Levels of FFD for Digital Logic with No Added Testability

Level of Assembly	Complexity (L, M, H)	Achieved FFD (%)	Cost (L, M, H)
Component	L	90	L
	M	77.5	L
	M	60	L
	M	84-89	L
	M	84-90	L
	N/A	60	L
Board	M	70-99	M/H
	M+/H	99.65	H
	M+/H	99.73	H
System	N/A	70	L

Table II-4. Achieved Levels of FFD for Digital Logic with DFT

Level of Assembly	**Complexity (L, M, H)**	**Achieved FFD (%)**	**DFT cost (L, M, H)**
Component	M	97.5	H
	H	98-100	H
	H	97.5	M
	H	90	M
Board	M+/H	99.9	H
	M+/H	99.9	H
System	N/A	70-90	M

Table II-5. Achieved Levels of FFD for Digital Logic with BIT

Level of Assembly	**Complexity (L, M, H)**	**Achieved FFD (%)**	**BIT Cost (L, M, H)**
Component	L	97.5	M+
	M	92.5	L/M
	M	95	M+
	M	95	M+
	H	97.5	M+
	H	85	M+
	H	96	M+
	H	60	M
Board	N/A	60-85	M
	N/A	85+	H
	H	95	M+
System	N/A	95	H

From this data, we observed that, as a general rule, the incorporation of BIT and/or DFT in a digital component, board, or system results in relatively high fault detection coverage (92+%). The exception to this rule-of-thumb is the BIT employed in COTS may or may not provide a high level of fault detection capability to the eventual user of the chip. This can probably be explained by the fact that a person purchasing a COTS chip for use will have limited information on the physical design of the component, much less on any detailed test features that might be used to increase coverage. Further, the market, to date, for components that are capable of passing test data upward through a hierarchy in which they are used, has been limited, although this is changing with the increased use of boundary scan across the industry.

A second conclusion from this data is that it is very important to consider the complexity of a component in deciding whether or not to implement DFT and BIT. If component testing is the sole goal for a low-complexity chip, it may not be cost effective to incorporate test features. On the other hand, the use of even low-complexity components in long life-time systems probably necessitates the application of DFT or BIT techniques to allow for the application and upward propagation of test data in the system environment.

II-1.2 FFI

The data above can also be used to estimate practical TFOM levels for various levels of assembly and various levels of complexity. The data pertaining to FFI is summarized in Table II-6.

Table II-6. Achieved Levels of FFI for Digital Logic

Level of Assembly	**Complexity (L, M, H)**	**Achieved FFI (%)**	**Cost (L, M, H)**
Board	M+	95+	M+
Board/System	M+	60	L
	M+	60-85	M
	M+	85+	H
	N/A	90+	M+/H
System	H	95	H+

From this data, we again established low, medium, and high levels for FFI, relative to the cost of attaining those levels, the level of assembly, and (for components) the complexity of the design. Based on the data collected, the FFI relations shown in Table II-7 were developed.

Table II-7. Practical Levels of FFI for Digital Logic

Level of Assembly	**Complexity (L, M, H)**	**Low Cost FFI (%)**	**Medium Cost FFI (%)**	**High Cost FFI (%)**
Board	N/A	<65	65-90	90+
System	N/A	<60	60-85	85-90+

We again broke the cost data out further into costs incurred for components and systems implementing DFT, components and systems implementing BIT (possibly including DFT), and components and systems tested by conventional means. Divided along these lines, we obtained the breakdown shown in Tables II-8 through II-10.

Table II-8. Achieved Levels of FFI for Digital Logic with No Added Testability

Level of Assembly	**Complexity (L, M, H)**	**Achieved FFI (%)**	**Cost (L, M, H)**
Board	L	60	L

Table II-9. Achieved Levels of FFI for Digital Logic with DFT

Level of Assembly	**Complexity (L, M, H)**	**Achieved FFI (%)**	**Cost (L, M, H)**
Board	M+	95+	M+
Board/System	M+	60-85	M
	M+	85+	H

Table II-10. Achieved Levels of FFI for Digital Logic with BIT

Level of Assembly	**Complexity (L, M, H)**	**Achieved FFI (%)**	**Cost (L, M, H)**
Board/System	N/A	90+	M+/H
System	H	95	H+

For FFI, the efficacy of using DFT and BIT, despite their cost, is even more apparent than for FFD. In all cases, a reasonable level of FFI for boards and system is unattainable without the incorporation of DFT or BIT structures.

II-1.3 FFA

The F16A [29] demonstrates the trade-offs between FFD/FFI and FFA. FFD achieved is 95%, but CND rates of 31% and RTOK rates of 22% are experienced. This scenario can result from insufficient fault isolation capabilities, in which case the failure condition cannot be properly identified, or from tests that are very stringent to achieve a high probability of fault detection and fault isolation, but at a cost of susceptibility to transients and environmental conditions.

Williams and Hawkins have published an extensive theoretical analysis on the probability of false alarms [30]. They note that false alarms are actually analogous to Type 1 statistical errors (the probability that a test declares a good part faulty). A survey of production tests found that as many as 30-50% of failed parts are actually good, indicating a higher than expected rate of false alarms for even production tests. Using probabilistic analysis, Hawkins and Williams demonstrate that

$$Prob.\ of\ false\ alarm = \frac{\alpha \ast Yield}{1 - Yield^{FFD} + \alpha \ast Yield}$$

where α is a parameter defining the confidence of the test employed and "yield" is the yield of the particular test. Suppose a system has a yield of 60% and FFD of 100%. Under these conditions, to achieve a probability of a false alarm of 5%, statistically, the confidence required in the test is 96.5%, which, as the authors point out, is probably impossible, particularly for BIT. Moreover, as components/modules/systems move further along in the life-cycle, the yield of the tests performed is expected to be higher, which actually increases the level of confidence required for the particular test. The key, then, may be to focus on achieving the levels of FFD and FFI (or other fault detection and isolation metrics) required, and then looking for techniques to minimize the probability of false alarms.

II-1.4 T_D/T_I

There is insufficient data to determine practically achieved levels of T_D and/or T_I. Intuitively, T_D will increase as the level and cost of FFD increases, reflecting either a greater number of required test patterns or a greater use of BIT (which typically requires a longer execution cycle due to the use of random or pseudo-random versus deterministic patterns). On the other hand, T_I can be expected to decrease significantly as the level of FFI increases since a greater percentage of fault isolation time can be expected to be expended to "troubleshoot" faults that are not isolated through the use of external test equipment (ETE) or BIT (which are fewer as FFI increases). For the same level of FFI, it is also intuitive that BIT techniques actually decrease the time required for fault isolation, because much of the time required for fault isolation is consumed in manual operations.

II-2. Practical Levels of Analog/Mixed Signal Circuit TFOMs

In general, based on historical industry data, TFOM requirement values for analog and mixed signal circuits have been less severe than for digital circuits, primarily because of the lack of advanced test technology and theoretical foundations for that technology in the analog domain.

For example, achieved fault detection coverage requirements are typically anywhere from 5% to 20% lower than those for digital circuits, thus often falling in the 60 - 80% range. Fault isolation ambiguity group sizes are often two to five times the sizes for digital circuits. For example, it is not unusual to see ambiguity groups of 10 to 15 components specified for analog circuits, versus a typical 2 to 4 components for digital circuits.

However, it is important not to draw too many conclusions from such generic industry data for three major reasons:

a. It is often not known what fault model was used or assumed.

b. It is often not known what means (and accuracy of same) was used for predicting, verifying, or measuring the data.

c. Advancements in analog test technology (particularly work on structured analog DFT techniques and IEEE P1149.4), tremendously impact the practical levels of TFOMs that can be achieved, just as internal scan and IEEE 1149.1 have dramatically impacted the practical levels of TFOMs for digital circuits.

It is very difficult to make a generalization that at a specific value of fault detection, isolation, or test time that the negative impact on the above factors increases sharply. Several statements can be made, however:

a. The point of sharp penalty increase can be different for different implementations of a given mixed signal function (e.g., amount of digital circuitry used versus analog).

b. The point of sharp penalty increase can vary for the same implementation versus the frequency of operation (i.e., low frequency versus high frequency versus RF/microwave).

c. The point of sharp penalty increase can be different for different circuit characteristics, such as signal levels, power levels, carrier frequency, timing circuit frequency, pulse characteristics, bandwidth, dynamic range, and whether frequency agility and coherency are requirements.

d. In general, a sharp increase in penalties occurs in the transition area between high frequencies and microwave frequencies (between approximately 100 MHz to 500 MHz).

In short, it is almost impossible to make quantitative generalizations in regard to practically achieved levels of TFOMs for analog and mixed signal circuits, if only because their design, operational, and input parameters can vary so widely. One option might be to group analog and mixed signal designs based on those parameters and look for trends; however, the amount of data necessary for this type of analysis on analog and/or mixed signal circuits is simply not available.

On the other hand, what we can do is make qualitative assessments that can be used as design "monitors," indications that achieving testability may become exponentially more expensive if certain design parameters are adopted. An example might be a red flag on a design scheme that utilizes all RF components versus another scheme that utilizes mostly low frequency analog components. The general observations laid out above are a step in this direction.

The data presented in Tasks 1 and 2 were combined to provide an estimation of practical levels of TFOM values for various levels of assembly of analog and mixed signal systems.

II-2.1 FFD

The data from tasks 1 and 2 related to FFD is summarized in the table below.

Table II-11. Achieved FFD for Analog and Mixed Signal Circuits

Case	Circuit Type	Frequency	Complexity	FFD	Cost
C1	mixed sig	low (<5 KHz)	high	? (BIT)	low
B1	analog	low (DC)	low	67%, 76% (BIT, BIT + Manual Proc)	medium
B2	analog	low (DC)	low	100%, 100% (BIT, BIT + Manual Proc)	medium
B3	analog	low (DC)	low	93%, 95% (BIT, BIT + Manual Proc)	medium
B4	mixed sig	mixed (high to RF)	high	97% (BIT)	high
S1	mixed sig	mixed (high to RF)	NA	92% (BIT)	high
S2	mixed sig	mixed (low to RF)	NA	80% (est.) (BIT)	medium
S3	mixed sig	mixed (low to RF)	NA	93.4% (BIT)	high
S4	mixed sig	mixed (low to RF)	NA	98%+ (BIT)	high
S5	mixed sig	mixed (low to RF)	NA	67% (BIT)	medium
S6	mixed sig	mixed (low to RF)	NA	94.5% (BIT)	high

Based on the data in the table above, the data presented in the digital section of the report, and additional industry data associated with proprietary, commercial analog and RF systems, the practical levels of fault detection coverage are presented in the tables below. The data is broken down by analog, versus mixed signal circuits; by frequency range; and by low, medium, and high cost. The abbreviations for frequency range are as follows:

L - low frequency operation (less than 1 MHz)
H - high frequency operation (1 MHz to 100 MHz)
V - very high frequency (greater than 100 MHz)

The breakdown of the data by frequency was chosen because frequency was found to be a factor that greatly affects the breakpoints for penalties (cost, performance degradation, etc.) for various levels of TFOM values. Complexity, another factor shown in the table above, was found to be too subjective to quantify, particularly for the system level.

Table II-12. Practical Levels of FFD for Analog Circuits - Low Cost

Assembly Level	**V**	**H**	**L**
Component	50 - 65	55 - 70	65 - 80
Board	<50	<55	<60
System	<45	<50	<55

Table II-13. Practical Levels of FFD for Analog Circuits - Medium Cost

Assembly Level	**V**	**H**	**L**
Component	75 - 80	78 - 85	80 - 87
Board	50 - 70	65 - 75	70 - 80
System	50 - 65	60 - 70	68 - 75

Table II-14. Practical Levels of FFD for Analog Circuits - High Cost

Assembly Level	**V**	**H**	**L**
Component	78 - 83	80 - 85	83 - 88+
Board	70 - 78	75 - 82	80 - 85+
System	65 - 73	70 - 77	75 - 80+

Table II-15. Practical Levels of FFD for Mixed Signal Circuits - Low Cost

Assembly Level	**V**	**H**	**L**
Component	55 - 70	60 - 75	70 - 85
Board	<55	<60	<65
System	<50	<55	<60

Table II-16. Practical Levels of FFD for Mixed Signal Circuits - Medium Cost

Assembly Level	**V**	**H**	**L**
Component	80 - 85	83 - 90	85 - 92
Board	55 - 75	70 - 80	75 - 85
System	55 - 70	65 - 75	73 - 80

Table II-17. Practical Levels of FFD for Mixed Signal Circuits - High Cost

Assembly Level	**V**	**H**	**L**
Component	83 - 88	85 - 90	88 - 93+
Board	75 - 83	80 - 87	85 - 90+
System	70 - 78	75 - 82	80 - 85+

II-2.2 FFI

The data from tasks 1 and 2 related to FFI is summarized in the table below.

Table II-18. Achieved FFI for Analog and Mixed Signal Circuits

Case	Circuit Type	Frequency	Complexity	FFI	Cost
C1	mixed sig	low (<5 KHz)	high	? (BIT)	low
B1	analog	low (DC)	low	?	medium
B2	analog	low (DC)	low	?	medium
B3	analog	low (DC)	low	?	medium
B4	mixed sig	mixed (high to RF)	high	85% (BIT) 90% (B/T) 99% (BIT/Tester/Manual Procedures)	high
S1	mixed sig	mixed (high to RF)	NA	97.2% 1 LRU 59% 1 SRU 0% 2 SRU 100% 3 SRU (BIT)	high
S2	mixed sig	mixed (low to RF)	NA	? (BIT)	medium
S3	mixed sig	mixed (low to RF)	NA	74.7 1 LRU (BIT)	high
S4	mixed sig	mixed (low to RF)	NA	96% 1 LRU (BIT)	high
S5	mixed sig	mixed (low to RF)	NA	80% 1 LRU (BIT)	medium
S6	mixed sig	mixed (low to RF)	NA	73.5% 1 LRU (BIT)	high

Based on the data in the table above, the data presented in the digital section of the report, and additional industry data associated with proprietary, commercial analog and RF systems, the practical levels of fault isolation coverage are presented in the tables below. The data is broken down by analog versus mixed signal circuits, by frequency range, and by low, medium, and high cost.

Table II-19. Practical Levels of FFI for Analog Circuits - Low Cost

Assembly Level	**V**	**H**	**L**
Board	<40	<45	<50
System	<35	<40	<45

Table II-20. Practical Levels of FFI for Analog Circuits - Medium Cost

Assembly Level	**V**	**H**	**L**
Board	50 - 65	60 - 75	70 - 80
System	45 - 60	55 - 70	65 - 75

Table II-21. Practical Levels of FFI for Analog Circuits - High Cost

Assembly Level	**V**	**H**	**L**
Board	65 - 75	70 - 80	75 - 85+
System	60 - 70	65 - 75	70 - 80+

Table II-22. Practical Levels of FFI for Mixed Signal Circuits - Low Cost

Assembly Level	**V**	**H**	**L**
Board	<45	<50	<55
System	<40	<45	<50

Table II-23. Practical Levels of FFI for Mixed Signal Circuits - Medium Cost

Assembly Level	**V**	**H**	**L**
Board	55 - 70	65 - 80	75 - 85
System	50 - 65	60 - 75	70 - 80

Table II-24. Practical Levels of FFI for Mixed Signal Circuits - High Cost

Assembly Level	**V**	**H**	**L**
Board	70 - 80	75 - 85	80 - 90+
System	65 - 75	70 - 80	75 - 85+

Appendix III. DFT/BIT Technique Scoreboard Tables

The evaluation of DFT/BIT techniques described in Section 3 resulted in the following scoreboard tables. The techniques were grouped into fourteen categories, each of which is covered in a section of this appendix. The criteria used to evaluate the techniques vary by technique category, with the basic fourteen criteria as follows:

1. Area overhead
2. I/O overhead
3. Power overhead
4. Computational Expense of Design Translation
5. Performance Degradation
6. Test Run Time
7. Stuck-at Fault Coverage
8. Delay/Open/Bridging Fault Coverage
9. Impact on Availability, Reliability, Maintainability
10. Applicability to Various Circuit Structures
11. Compatibility With Other DFT/BIT Structures
12. Compatibility With IEEE Standard 1149.1 and Other Test Buses
13. ATE/Diagnostics Accessibility
14. Life Cycle Usefulness

Changes to the applied criteria are listed at the beginning of the categories' scoreboards.

Each technique was given a numerical score for each criterion applicable to that technique category. The scores represent relative benefits of implementing the techniques. A higher score indicates higher benefit. The maximum scores assigned varied between the criteria. The minimum score assigned was zero. Scores were assigned for three different life-cycle phases (development, manufacturing, and field) over three application environments (ground-based, airborne, and spaceborne).

III-1. Structured Chip-level BIT Techniques

Concurrent Built-In Self-Test Architecture (CBIST)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	0	0	0	0	0	0	0	0	0
4	10	0	0	10	0	0	10	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	0	10	10	0	10	15	0	10	20
Sub	57.5	62.5	77.5	62.5	62.5	80.0	60.0	62.5	77.5
Total	197.5			205.0			200.0		

Cyclic Analysis Testing System (CATS)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	5	5	15	10	5	20	15	5	25
4	20	0	0	20	0	0	20	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5

7	12	12	12	12	12	8	8	12	4
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	0	10	10	0	10	15	0	10	20
Sub	82.0	87.0	102.0	97.0	87.0	115.5	105.5	87.0	124.0
Total			271.0			299.5			316.5

Circular Built-In Self-Test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	12	12	12	12	12	8	8	12	4
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	5	10	15	5	10	15	5	10	5
14	0	10	10	0	10	15	0	10	20
Sub	54.5	77.0	92.0	64.5	77.0	100.5	68.0	77.0	99.0
Total			223.5			242.0			244.0

Circular Self-Test Path (CSTP)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	12	12	12	12	12	8	8	12	4
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	0	0	10	0	0	15	0	0	20
Sub	52.0	62.0	84.5	62.0	62.0	93.0	65.5	62.0	96.5
Total	198.5			217.0			224.0		

Built-In Logic Block Observation (BILBO)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	0	0	0	0	0	0	0	0	0
4	10	0	0	10	0	0	10	0	0
5	0	0	0	0	0	0	0	0	0
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5

8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	5	10	15	5	10	15	5	10	5
14	0	10	10	0	10	15	0	10	20
Sub	57.5	65.0	77.5	62.5	65.0	80.0	60.0	65.0	72.5
Total			200.0			207.5			197.5

Built-In Exhaustive Self Test (BEST)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	5	10	15	5	10	15	5	10	5
14	0	10	10	0	10	15	0	10	20
Sub	42.5	67.5	85.0	47.5	67.5	87.5	45.0	67.5	80.0
Total			195.0			202.5			192.5

Design Methodology for Incorporating Self-Test (DEMIST)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	15	0	0	15	0	0	15	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	5	10	15	5	10	15	5	10	5
14	0	10	10	0	10	15	0	10	20
Sub	55.0	65.0	77.5	60.0	65.0	80.0	57.5	65.0	72.5
Total	197.5			205.0			195.0		

Level-Sensitive Scan Design On-Chip Self-Test (LOCST)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	0	0	0	0	0	0	0	0	0
7	15	15	15	15	15	10	10	15	5

8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	75.0	72.5	87.5	85.0	72.5	95.0	87.5	72.5	92.5
Total	235.0			252.5			252.5		

Simultaneous Self-Test (SST)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	12	12	12	12	12	8	8	12	4
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	0	10	10	0	10	15	0	10	20
Sub	64.5	79.5	92.0	74.5	79.5	100.5	78.0	79.5	104.0
Total	236.0			254.5			261.5		

International Computers Limited Built-In Self-Test (ICL BIST)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	15	0	0	15	0	0	15	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	0	0	0	0	0	0	0	0	0
13	5	10	15	5	10	15	5	10	5
14	0	10	10	0	10	15	0	10	20
Sub	52.5	57.5	82.5	55.0	57.5	82.5	50.0	57.5	72.5
Total	192.5			195.0			180.0		

On-Chip ROM

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	10	15	5	15	15	10	20	15	15
3	0	0	0	0	0	0	0	0	0
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7	15	15	15	15	15	10	10	15	5

8	10	10	15	10	10	10	5	10	5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	0	10	10	0	10	15	0	10	20
Sub	72.5	85.0	102.5	80.0	85.0	105.0	77.5	85.0	102.5
Total	260.0			270.0			265.0		

Microprocessor Built-In Test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	10	15	5	15	15	10	20	15	15
3	5	5	15	10	5	20	15	5	25
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	10	10	15	10	10	10	5	10	5
9	5	10	15	10	10	20	15	10	25
10	0	0	0	0	0	0	0	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	92.5	112.5	130.0	112.5	112.5	145.0	122.5	112.5	150.0
Total	335.0			370.0			385.0		

III-2. Structured Chip-level DFT Techniques

Level-Sensitive Scan Design (LSSD)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	5	10	15	10	10	20	15	10	25
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	5	10	15	5	10	15	5	10	5
14	10	0	10	10	0	15	10	0	20
Sub	85.0	77.5	115.0	97.5	77.5	125.0	102.5	77.5	125.0
Total	277.5			300.0			305.0		

Weighted Random Test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5

7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	10	10	0	10	10	0	10	10	0
Sub	80.0	77.5	90.0	90.0	77.5	92.5	92.5	77.5	90.0
Total	247.5			260.0			260.0		

Pseudo-Exhaustive Test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	5	5	15	10	5	20	15	5	25
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	15	0	0	15	0	0	15	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	10	10	10	10	10	15	10	10	20
Sub	87.5	80.0	102.5	100.0	80.0	112.5	105.0	80.0	117.5
Total	270.0			292.5			302.5		

CrossCheck

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	5	5	15	10	5	20	15	5	25
4	0	0	0	0	0	0	0	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	5	10	15	10	10	20	15	10	25
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	80.0	95.0	130.0	95.0	95.0	142.5	102.5	95.0	145.0
Total	305.0			332.5			342.5		

Quiescent Power Supply Current Test (IDDQ Test)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	15	15	15	15	15	10	10	15	5

8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	5	0	10	5	0	10	5	0	10
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	0	10	10	0	10	15	0	10	20
Sub	62.5	82.5	100.0	72.5	82.5	107.5	75.0	82.5	110.0
Total	245.0			262.5			267.5		

III-3. Ad-hoc DFT Techniques

Criteria additions:

7d Impact on Fault Coverage/Test Pattern Generation Time

Provide initialization capability

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	10	15	5	15	15	10	20	15	15
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
7d	15	10	10	10	10	10	5	10	5
Sub	55.0	50.0	35.0	60.0	50.0	45.0	65.0	50.0	50.0
Total	140.0			155.0			165.0		

Provide global feedback path control

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	10	15	5	15	15	10	20	15	15
4	15	0	0	15	0	0	15	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7d	7.5	5	5	5	5	5	2.5	5	2.5
Sub	45.0	42.5	22.5	52.5	42.5	32.5	60.0	42.5	40.0
Total	110.0			127.5			142.5		

Provide partioning of large circuits

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7d	15	10	10	10	10	10	5	10	5
Sub	35.0	30.0	22.5	35.0	30.0	27.5	35.0	30.0	27.5
Total			87.5			92.5			92.5

Add test points to enhance controllability and observability

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	0	0	0	0	0	0	0	0	0
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7d	15	10	10	10	10	10	5	10	5
Sub	30.0	22.5	20.0	27.5	22.5	22.5	25.0	22.5	20.0
Total			72.5			72.5			67.5

Test embedded structures separately

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7d	15	10	10	10	10	10	5	10	5
Sub	22.5	20.0	20.0	20.0	20.0	22.5	17.5	20.0	20.0
Total	62.5			62.5			57.5		

Separate mixed logic families

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	10	15	5	15	15	10	20	15	15
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7d	15	10	10	10	10	10	5	10	5
Sub	40.0	37.5	25.0	42.5	37.5	32.5	45.0	37.5	35.0
Total	102.5			112.5			117.5		

Limit chip fanout at test points to one less than maximum

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	10	15	5	15	15	10	20	15	15
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
7d	7.5	5	5	5	5	5	2.5	5	2.5
Sub	47.5	45.0	30.0	55.0	45.0	40.0	62.5	45.0	47.5
Total			122.5			140.0			155.0

Disable free running system clocks during test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	10	15	5	15	15	10	20	15	15
4	15	0	0	15	0	0	15	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7d	15	10	10	10	10	10	5	10	5
Sub	52.5	47.5	27.5	57.5	47.5	37.5	62.5	47.5	42.5
Total			127.5			142.5			152.5

Disable internal one-shots during test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	10	15	5	15	15	10	20	15	15
4	15	0	0	15	0	0	15	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7d	7.5	5	5	5	5	5	2.5	5	2.5
Sub	45.0	42.5	22.5	52.5	42.5	32.5	60.0	42.5	40.0
Total	110.0			127.5			142.5		

Avoid redundant logic

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	10	15	5	15	15	10	20	15	15
4	15	0	0	15	0	0	15	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7d	15	10	10	10	10	10	5	10	5
Sub	52.5	47.5	27.5	57.5	47.5	37.5	62.5	47.5	42.5
Total	127.5			142.5			152.5		

Avoid wired AND/OR functions

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	10	15	5	15	15	10	20	15	15
4	7.5	0	0	7.5	0	0	7.5	0	0
5	5	5	15	5	5	15	5	5	15
7d	15	10	10	10	10	10	5	10	5
Sub	42.5	40.0	32.5	45.0	40.0	40.0	47.5	40.0	42.5
Total			115.0			125.0			130.0

Avoid asynchronous design practices

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	10	15	5	15	15	10	20	15	15
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7d	15	10	10	10	10	10	5	10	5
Sub	45.0	47.5	27.5	50.0	47.5	37.5	55.0	47.5	42.5
Total			120.0			135.0			145.0

Avoid clock gating

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	10	15	5	15	15	10	20	15	15
4	0	0	0	0	0	0	0	0	0
5	5	5	15	5	5	15	5	5	15
7d	15	10	10	10	10	10	5	10	5
Sub	35.0	40.0	32.5	37.5	40.0	40.0	40.0	40.0	42.5
Total	107.5			117.5			122.5		

III-4. RAM BIT Techniques

Criteria modifications:

7a Cell Stuck-at Fault Coverage
8a Cell Coupling Fault Coverage
8b Decoder Fault Coverage

Built-In Self-Test for Embedded Static RAMs

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
4	10.5	0	0	10.5	0	0	10.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7a	10	10	10	5	10	5	10	10	10
8a	5	5	5	4	5	4	3	5	3
8b	5	5	5	4	5	4	3	5	3
14	0	10	10	0	10	15	0	10	20
Sub	35.5	40.0	45.0	28.5	40.0	43.0	31.5	40.0	51.0
Total	120.5			111.5			122.5		

Bardell and McAnney's Test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	3	6	1.5	4.5	6	3	6	6	4.5
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	3.5	10.5	10.5	3.5	10.5	10.5	3.5	10.5	10.5
7a	10	10	10	5	10	5	10	10	10
8a	0	0	0	0	0	0	0	0	0
8b	5	5	5	4	5	4	3	5	3
14	10	10	10	10	10	15	10	10	20
Sub	41.5	44.0	44.5	37.0	44.0	45.0	42.5	44.0	55.5
Total	130.0			126.0			142.0		

Parallel Test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
4	4.5	0	0	4.5	0	0	4.5	0	0
5	4	4	12	4	4	12	4	4	12
6	5	15	15	5	15	15	5	15	15
7a	10	10	10	5	10	5	10	10	10
8a	0	0	0	0	0	0	0	0	0
8b	5	5	5	4	5	4	3	5	3
14	10	10	0	10	10	0	10	10	0
Sub	43.5	54.0	44.5	40.0	54.0	41.0	46.5	54.0	47.5
Total	142.0			135.0			148.0		

Mazumder and Patel's Test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
4	4.5	0	0	4.5	0	0	4.5	0	0
5	4	4	12	4	4	12	4	4	12
6	5	15	15	5	15	15	5	15	15
7a	10	10	10	5	10	5	10	10	10
8a	2.5	2.5	2.5	2	2.5	2	1.5	2.5	1.5
8b	5	5	5	4	5	4	3	5	3
14	10	10	0	10	10	0	10	10	0
Sub	46.0	56.5	47.0	42.0	56.5	43.0	48.0	56.5	49.0
Total			149.5			141.5			153.5

Exhaustive Random Sequence (bidirectional)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
4	12	0	0	12	0	0	12	0	0
5	5	5	15	5	5	15	5	5	15
6	3.5	10.5	10.5	3.5	10.5	10.5	3.5	10.5	10.5
7a	10	10	10	5	10	5	10	10	10
8a	0	0	0	0	0	0	0	0	0
8b	5	5	5	4	5	4	3	5	3
14	10	10	5	10	10	7.5	10	10	10
Sub	50.5	50.5	48.0	47.0	50.5	47.0	53.5	50.5	56.0
Total			149.0			144.5			160.0

Exhaustive Random Sequence (unidirectional)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	8	16	4	12	16	8	16	16	12
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
6	3.5	10.5	10.5	3.5	10.5	10.5	3.5	10.5	10.5
7a	10	10	10	5	10	5	10	10	10
8a	0	0	0	0	0	0	0	0	0
8b	4	4	4	3.2	4	3.2	2.4	4	2.4
14	10	10	5	10	10	7.5	10	10	10
Sub	55.5	55.5	48.5	53.7	55.5	49.2	61.9	55.5	59.9
Total	159.5			158.4			177.3		

Exhaustive Random Sequence (random checker)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7a	10	10	10	5	10	5	10	10	10
8a	0	0	0	0	0	0	0	0	0
8b	4	4	4	3.2	4	3.2	2.4	4	2.4
14	10	10	5	10	10	7.5	10	10	10
Sub	56.5	56.5	46.5	55.7	56.5	48.2	64.9	56.5	59.9
Total	159.5			160.4			181.3		

Walking 1/0 Test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
6	0	0	0	0	0	0	0	0	0
7a	10	10	10	5	10	5	10	10	10
8a	5	5	5	4	5	4	3	5	3
8b	5	5	5	4	5	4	3	5	3
14	10	0	0	10	0	0	10	0	0
Sub	60.0	45.0	40.0	58.0	45.0	38.0	66.0	45.0	46.0
Total	145.0			141.0			157.0		

MATS++

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
6	3.5	10.5	10.5	3.5	10.5	10.5	3.5	10.5	10.5
7a	10	10	10	5	10	5	10	10	10
8a	0	0	0	0	0	0	0	0	0
8b	5	5	5	4	5	4	3	5	3
14	10	10	0	10	10	0	10	10	0
Sub	58.5	60.5	45.5	57.5	60.5	44.5	66.5	60.5	53.5
Total	164.5			162.5			180.5		

March C-

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7a	10	10	10	5	10	5	10	10	10
8a	2.5	2.5	2.5	2	2.5	2	1.5	2.5	1.5
8b	5	5	5	4	5	4	3	5	3
14	10	10	0	10	10	0	10	10	0
Sub	60.0	60.0	45.0	58.5	60.0	43.5	67.0	60.0	52.0
Total			165.0			162.0			179.0

Jacobson's Test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
6	3.5	10.5	10.5	3.5	10.5	10.5	3.5	10.5	10.5
7a	10	10	10	5	10	5	10	10	10
8a	0	0	0	0	0	0	0	0	0
8b	5	5	5	4	5	4	3	5	3
14	10	10	0	10	10	0	10	10	0
Sub	58.5	60.5	45.5	57.5	60.5	44.5	66.5	60.5	53.5
Total			164.5			162.5			180.5

III-5. ROM BIT Techniques

Criteria modifications:
7b Fault Coverage

Count-Based Compaction (checksums)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
4	10.5	0	0	10.5	0	0	10.5	0	0
6	5	15	15	5	15	15	5	15	15
7b	7.5	7.5	7.5	5	7.5	5	2.5	7.5	2.5
Sub	28.0	32.5	25.0	28.0	32.5	25.0	28.0	32.5	25.0
Total			85.5			85.5			85.5

Parity-Based Compaction

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	8	16	4	12	16	8	16	16	12
4	10.5	0	0	10.5	0	0	10.5	0	0
6	5	15	15	5	15	15	5	15	15
7b	4.5	4.5	4.5	3	4.5	3	1.5	4.5	1.5
Sub	28.0	35.5	23.5	30.5	35.5	26.0	33.0	35.5	28.5
Total			87.0			92.0			97.0

Polynomial Division-Based Compaction (signatures)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	8	16	4	12	16	8	16	16	12
4	10.5	0	0	10.5	0	0	10.5	0	0
6	5	15	15	5	15	15	5	15	15
7b	7.5	7.5	7.5	5	7.5	5	2.5	7.5	2.5
Sub	31.0	38.5	26.5	32.5	38.5	28.0	34.0	38.5	29.5
Total			96.0			99.0			102.0

Exhaustive Enhanced Output Data Modification (EEODM)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
4	7.5	0	0	7.5	0	0	7.5	0	0
6	4	12	12	4	12	12	4	12	12
7b	15	15	15	10	15	10	5	15	5
Sub	31.5	37.0	29.5	29.0	37.0	27.0	26.5	37.0	24.5
Total			98.0			93.0			88.0

III-6. PLA BIT Techniques

Criteria modifications:

7c Single Stuck-at Fault Coverage
8c Single Crosspoint Fault Coverage
8d Single Bridging Fault Coverage
8e Multiple Fault Coverage

Concurrent Test Using Error Detection Code

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	3	6	1.5	4.5	6	3	6	6	4.5
2	6	9	3	9	9	6	12	9	9
4	0	0	0	0	0	0	0	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7c	10	10	10	8	10	8	5	10	5
8b	5	5	5	4	5	4	2.5	5	2.5
8c	0	0	0	0	0	0	0	0	0
8d	0	0	0	0	0	0	0	0	0
9	5	10	15	10	10	20	15	10	25
Sub	39.0	60.0	64.5	45.5	60.0	71.0	50.5	60.0	76.0
Total	163.5			176.5			186.5		

Design with Cumulative Parity Comparison

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	2	3	1	3	3	2	4	3	3
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7c	10	10	10	8	10	8	5	10	5
8b	5	5	5	4	5	4	2.5	5	2.5
8c	5	5	5	4	5	4	2.5	5	2.5
8d	5	5	5	4	5	4	2.5	5	2.5
9	1.5	3	4.5	3	3	6	4.5	3	7.5
Sub	48.5	58.5	55.5	48.5	58.5	55.5	46.0	58.5	53.0
Total			162.5			162.5			157.5

Divide and Conquer Strategy

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	2	3	1	3	3	2	4	3	3
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7c	10	10	10	8	10	8	5	10	5
8b	5	5	5	4	5	4	2.5	5	2.5
8c	0	0	0	0	0	0	0	0	0
8d	5	5	5	4	5	4	2.5	5	2.5
9	1.5	3	4.5	3	3	6	4.5	3	7.5
Sub	41.0	46.0	43.0	42.0	46.0	44.0	41.0	46.0	43.0
Total			130.0			132.0			130.0

Test with Low Overhead and High Fault Coverage

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	7	14	3.5	10.5	14	7	14	14	10.5
2	4	6	2	6	6	4	8	6	6
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7c	10	10	10	8	10	8	5	10	5
8b	5	5	5	4	5	4	2.5	5	2.5
8c	0	0	0	0	0	0	0	0	0
8d	5	5	5	4	5	4	2.5	5	2.5
9	1.5	3	4.5	3	3	6	4.5	3	7.5
Sub	47.5	60.5	52.5	50.5	60.5	55.5	51.5	60.5	56.5
Total	160.5			166.5			168.5		

Exhaustive Test with Signature Analysis

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	7	14	3.5	10.5	14	7	14	14	10.5
2	6	9	3	9	9	6	12	9	9
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
6	0	0	0	0	0	0	0	0	0
7c	10	10	10	8	10	8	5	10	5
8b	5	5	5	4	5	4	2.5	5	2.5
8c	5	5	5	4	5	4	2.5	5	2.5
8d	5	5	5	4	5	4	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
Sub	60.5	58.0	54.0	64.5	58.0	58.0	66.0	58.0	59.5
Total	172.5			180.5			183.5		

Built-In Self-Test Input Generator (BISTIG)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	6	9	3	9	9	6	12	9	9
4	12	0	0	12	0	0	12	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7c	10	10	10	8	10	8	5	10	5
8b	5	5	5	4	5	4	2.5	5	2.5
8c	5	5	5	4	5	4	2.5	5	2.5
8d	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
Sub	50.5	54.0	48.0	54.5	54.0	52.0	56.5	54.0	54.0
Total	152.5			160.5			164.5		

Self-Checking Design Using Alternating Logic

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	4	6	2	6	6	4	8	6	6
4	15	0	0	15	0	0	15	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7c	10	10	10	8	10	8	5	10	5
8b	5	5	5	4	5	4	2.5	5	2.5
8c	5	5	5	4	5	4	2.5	5	2.5
8d	0	0	0	0	0	0	0	0	0
9	5	10	15	10	10	20	15	10	25
Sub	56.5	63.5	62.0	62.0	63.5	67.5	65.5	63.5	71.0
Total	182.0			193.0			200.0		

III-7. Coding Techniques

Criteria modifications:

7e	Error Detection Capability
8f	Error Correction Capability

Cyclic Codes

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7e	15	15	15	15	15	15	15	15	15
8f	10	10	15	10	10	15	10	10	15
10	0	0	0	0	0	0	0	0	0
Sub	40.0	37.5	40.0	42.5	37.5	42.5	45.0	37.5	45.0
Total	117.5			122.5			127.5		

Hamming Codes

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7e	7.5	7.5	7.5	7.5	7.5	7.5	7.5	7.5	7.5
8f	10	10	15	10	10	15	10	10	15
10	7.5	0	0	7.5	0	0	7.5	0	0
Sub	40.0	30.0	32.5	42.5	30.0	35.0	45.0	30.0	37.5
Total	102.5			107.5			112.5		

Arithmetic Codes

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7e	15	15	15	15	15	15	15	15	15
8f	5	5	7.5	5	5	7.5	5	5	7.5
10	0	0	0	0	0	0	0	0	0
Sub	35.0	32.5	32.5	37.5	32.5	35.0	40.0	32.5	37.5
Total	100.0			105.0			110.0		

Checksums

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
4	7.5	0	0	7.5	0	0	7.5	0	0
5	5	5	15	5	5	15	5	5	15
7e	15	15	15	15	15	15	15	15	15
8f	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
Sub	32.5	30.0	32.5	35.0	30.0	35.0	37.5	30.0	37.5
Total	95.0			100.0			105.0		

Berger Codes

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7e	15	15	15	15	15	15	15	15	15
8f	0	0	0	0	0	0	0	0	0
10	7.5	0	0	7.5	0	0	7.5	0	0
Sub	37.5	27.5	25.0	40.0	27.5	27.5	42.5	27.5	30.0
Total	90.0			95.0			100.0		

Parity Codes

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
7e	0	0	0	0	0	0	0	0	0
8f	0	0	0	0	0	0	0	0	0
10	7.5	0	0	7.5	0	0	7.5	0	0
Sub	37.5	25.0	20.0	42.5	25.0	25.0	47.5	25.0	30.0
Total	82.5			92.5			102.5		

Duplication Codes

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
7e	15	15	15	15	15	15	15	15	15
8f	0	0	0	0	0	0	0	0	0
10	15	0	0	15	0	0	15	0	0
Sub	40.0	17.5	22.5	40.0	17.5	22.5	40.0	17.5	22.5
Total	80.0			80.0			80.0		

III-8. Fault Tolerant Techniques

Duplication with Comparison

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	15	0	0	15	0	0	15	0	0
5	4	4	12	4	4	12	4	4	12
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	15	0	0	15	0	0	15	0	0
Sub	44.0	21.5	29.5	51.5	21.5	37.0	59.0	21.5	44.5
Total	95.0			110.0			125.0		

Static Redundancy

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
3	0	0	0	0	0	0	0	0	0
4	7.5	0	0	7.5	0	0	7.5	0	0
5	5	5	15	5	5	15	5	5	15
9	5	10	15	10	10	20	15	10	25
10	15	0	0	15	0	0	15	0	0
Sub	37.5	25.0	32.5	45.0	25.0	40.0	52.5	25.0	47.5
Total	95.0			110.0			125.0		

Dynamic Redundancy (Operative Standby)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	2	4	1	3	4	2	4	4	3
3	0	0	0	0	0	0	0	0	0
4	7.5	0	0	7.5	0	0	7.5	0	0
5	4	4	12	4	4	12	4	4	12
9	4	8	12	8	8	16	12	8	20
10	15	0	0	15	0	0	15	0	0
Sub	32.5	16.0	25.0	37.5	16.0	30.0	42.5	16.0	35.0
Total			73.5			83.5			93.5

Dynamic Redundancy (Inoperative Standby)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	2	4	1	3	4	2	4	4	3
3	5	5	15	10	5	20	15	5	25
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	15	0	0	15	0	0	15	0	0
Sub	27.0	16.5	31.0	35.5	16.5	39.5	44.0	16.5	48.0
Total			74.5			91.5			108.5

Hybrid Redundancy

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	5	5	15	5	5	15	5	5	15
9	5	10	15	10	10	20	15	10	25
10	15	0	0	15	0	0	15	0	0
Sub	25.0	15.0	30.0	30.0	15.0	35.0	35.0	15.0	40.0
Total	70.0			80.0			90.0		

Time Redundancy

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
3	5	5	15	10	5	20	15	5	25
4	7.5	0	0	7.5	0	0	7.5	0	0
5	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	15	0	0	15	0	0	15	0	0
Sub	40.0	30.0	27.5	52.5	30.0	40.0	65.0	30.0	52.5
Total	97.5			122.5			147.5		

Differential Cascode Voltage Switch Design (DCVS)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	3	6	1.5	4.5	6	3	6	6	4.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	7.5	0	0	7.5	0	0	7.5	0	0
5	5	5	15	5	5	15	5	5	15
9	5	10	15	10	10	20	15	10	25
10	0	0	0	0	0	0	0	0	0
Sub	23.0	23.5	39.0	32.0	23.5	48.0	41.0	23.5	57.0
Total	85.5			103.5			121.5		

III-9. Board Level Techniques

Criteria modifications:

1a	Overhead
7f	Fault Detection Capability
8g	Fault Isolation Capability
11a	Compatibility with Chip-level DFT/BIT Structures

Functional Test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1a	15	20	15	20	20	20	25	20	25
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7f	2	2	2	3	2	3	4	2	4
8g	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
11a	2.5	2.5	5	2.5	2.5	5	2.5	2.5	5
13	0	0	0	0	0	0	0	0	0
14	0	10	0	0	10	0	0	10	0
Sub	44.5	52.0	52.0	53.0	52.0	60.5	61.5	52.0	69.0
Total	148.5			165.5			182.5		

On-Board PRPG/MISR

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1a	3	4	3	4	4	4	5	4	5
4	12	0	0	12	0	0	12	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	0	0	0	0	0	0	0	0	0
7f	8	8	8	12	8	12	16	8	16
8g	8	8	8	12	8	12	16	8	16
9	2.5	5	7.5	5	5	10	7.5	5	12.5
11a	2.5	2.5	5	2.5	2.5	5	2.5	2.5	5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	10	10	10	10	10	15	10	10	20
Sub	51.0	45.0	56.5	62.5	45.0	73.0	74.0	45.0	84.5
Total	152.5			180.5			203.5		

On-Board Integration of VLSI Chip BIT

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1a	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	5	5	15	5	5	15	5	5	15
6	0	0	0	0	0	0	0	0	0
7f	10	10	10	15	10	15	20	10	20
8g	8	8	8	12	8	12	16	8	16
9	2.5	5	7.5	5	5	10	7.5	5	12.5
11a	4	4	8	4	4	8	4	4	8
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	10	10	10	10	10	15	10	10	20
Sub	42.0	47.0	66.0	53.5	47.0	82.5	65.0	47.0	94.0
Total	155.0			183.0			206.0		

On-Board ROM

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1a	0	0	0	0	0	0	0	0	0
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	4	12	12	4	12	12	4	12	12
7f	10	10	10	15	10	15	20	10	20
8g	10	10	10	15	10	15	20	10	20
9	2.5	5	7.5	5	5	10	7.5	5	12.5
11a	2.5	2.5	5	2.5	2.5	5	2.5	2.5	5
13	3.5	7	10.5	3.5	7	10.5	3.5	7	3.5
14	10	10	10	10	10	15	10	10	20
Sub	52.5	59.0	72.5	65.0	59.0	90.0	77.5	59.0	100.5
Total	184.0			214.0			237.0		

In-Circuit Test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1a	15	20	15	20	20	20	25	20	25
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7f	5	5	5	7.5	5	7.5	10	5	10
8g	10	10	10	15	10	15	20	10	20
9	2.5	5	7.5	5	5	10	7.5	5	12.5
11a	2.5	2.5	5	2.5	2.5	5	2.5	2.5	5
13	3.5	7	10.5	3.5	7	10.5	3.5	7	3.5
14	10	10	0	10	10	0	10	10	0
Sub	73.5	79.5	83.0	88.5	79.5	98.0	103.5	79.5	106.0
Total	236.0			266.0			289.0		

IEEE Standard 1149.1

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1a	7.5	10	7.5	10	10	10	12.5	10	12.5
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7f	10	10	10	15	10	15	20	10	20
8g	10	10	10	15	10	15	20	10	20
9	5	10	15	10	10	20	15	10	25
11a	5	5	10	5	5	10	5	5	10
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	65.0	75.0	92.5	82.5	75.0	115.0	100.0	75.0	127.5
Total	232.5			272.5			302.5		

III-10. System Level Techniques

Criteria modifications:
11b Compatibility with Board-level DFT/BIT Structures

IEEE Proposed Standard 1149.5

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1a	7.5	10	7.5	10	10	10	12.5	10	12.5
4	7.5	0	0	7.5	0	0	7.5	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7f	10	10	10	15	10	15	20	10	20
8g	10	10	10	15	10	15	20	10	20
9	5	10	15	10	10	20	15	10	25
11b	5	5	10	5	5	10	5	5	10
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	70.0	85.0	107.5	87.5	85.0	130.0	105.0	85.0	142.5
Total	262.5			302.5			332.5		

IEEE Standard 488.1

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1a	7.5	10	7.5	10	10	10	12.5	10	12.5
4	15	0	0	15	0	0	15	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7f	5	5	5	7.5	5	7.5	10	5	10
8g	10	10	10	15	10	15	20	10	20
9	2.5	5	7.5	5	5	10	7.5	5	12.5
11a	5	5	10	5	5	10	5	5	10
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	70.0	75.0	95.0	82.5	75.0	112.5	95.0	75.0	120.0
Total	240.0			270.0			290.0		

III-11. Intelligent BIT Techniques

Criteria modifications:
1a Overhead

Integrated BIT

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1a	5	7.5	5	7.5	7.5	7.5	10	7.5	10
4	7.5	0	0	7.5	0	0	7.5	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
9	4	8	12	8	8	16	12	8	20
14	10	10	10	10	10	15	10	10	20
Sub	29.0	28.0	34.5	35.5	28.0	46.0	42.0	28.0	57.5
Total			91.5			109.5			127.5

Adaptive BIT

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1a	5	7.5	5	7.5	7.5	7.5	10	7.5	10
4	10.5	0	0	10.5	0	0	10.5	0	0
5	5	5	15	5	5	15	5	5	15
9	5	10	15	10	10	20	15	10	25
14	5	5	5	5	5	7.5	5	5	10
Sub	30.5	27.5	40.0	38.0	27.5	50.0	45.5	27.5	60.0
Total			98.0			115.5			133.0

Temporal Monitoring BIT

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1a	10	15	10	15	15	15	20	15	20
4	15	0	0	15	0	0	15	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
14	10	10	10	10	10	15	10	10	20
Sub	40.0	32.5	35.0	47.5	32.5	47.5	55.0	32.5	60.0
Total	107.5			127.5			147.5		

III-12. Analog BIT Techniques

For input stimulus generation, use constant (DC) voltage sources

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	60.0	75.0	100.0	67.5	75.0	107.5	70.0	75.0	105.0
Total			235.0			250.0			250.0

For input stimulus generation, use constant current sources

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15

7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10-	10	10	10	10	15	10	10	20
Sub	60.0	75.0	100.0	67.5	75.0	107.5	70.0	75.0	105.0
Total	235.0			250.0			250.0		

For input stimulus generation, use sine wave generators

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	50.0	75.0	95.0	57.5	75.0	102.5	60.0	75.0	100.0
Total	220.0			235.0			235.0		

For input stimulus generation, use square wave generators

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	50.0	75.0	95.0	57.5	75.0	102.5	60.0	75.0	100.0
Total	220.0			235.0			235.0		

For input stimulus generation, use triangle wave generators

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5

8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	50.0	75.0	95.0	57.5	75.0	102.5	60.0	75.0	100.0
Total	220.0			235.0			235.0		

For input stimulus generation, use sawtooth wave generators

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	50.0	75.0	95.0	57.5	75.0	102.5	60.0	75.0	100.0
Total	220.0			235.0			235.0		

For input stimulus generation, use monolithic waveform generators

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	47.5	70.0	87.5	55.0	70.0	95.0	57.5	70.0	92.5
Total	205.0			220.0			220.0		

For input stimulus generation, use noise diode sources

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	5	5	15	10	5	20	15	5	25
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	5	15	15	5	15	15	5	15	15
7	0	0	0	0	0	0	0	0	0

8	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	30.0	52.5	55.0	42.5	52.5	70.0	55.0	52.5	80.0
Total	137.5			165.0			187.5		

For input stimulus generation, use a pseudorandom pattern generator and D/A conversion

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	22.5	37.5	45.0	27.5	37.5	52.5	32.5	37.5	55.0
Total	105.0			117.5			125.0		

For input stimulus generation, use a digitized waveform and D/A conversion

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	25.0	42.5	52.5	30.0	42.5	60.0	35.0	42.5	62.5
Total	120.0			132.5			140.0		

For coupling a stimulus source to a circuit, use existing antennas

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	10	15	5	15	15	10	20	15	15
3	5	5	15	10	5	20	15	5	25
4	20	0	0	20	0	0	20	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7	0	0	0	0	0	0	0	0	0

8	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	60.0	65.0	72.5	75.0	65.0	90.0	90.0	65.0	102.5
Total	197.5			230.0			257.5		

For coupling a stimulus source to a circuit, use analog multiplexing

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	30.0	45.0	70.0	35.0	45.0	80.0	40.0	45.0	80.0
Total	145.0			160.0			165.0		

For coupling a stimulus source to a circuit, use RF relays

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	15.0	25.0	22.5	17.5	25.0	27.5	20.0	25.0	27.5
Total			62.5			70.0			72.5

For coupling a stimulus source to a circuit, use directional couplers

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	0	0	0	0	0	0	0	0	0
4	10	0	0	10	0	0	10	0	0
5	0	0	0	0	0	0	0	0	0
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	0	0	0	0	0	0	0	0	0

8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	30.0	35.0	25.0	35.0	35.0	32.5	40.0	35.0	35.0
Total	90.0			102.5			110.0		

For coupling a stimulus source to a circuit, use transformers

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	5	10	15	5	10	15	5	10	5
14	5	5	5	5	5	7.5	5	5	10
Sub	22.5	37.5	45.0	30.0	37.5	55.0	37.5	37.5	55.0
Total	105.0			122.5			130.0		

For output response evaluation, use level detectors

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	57.5	75.0	95.0	65.0	75.0	102.5	67.5	75.0	100.0
Total			227.5			242.5			242.5

For output response evaluation, use frequency detectors

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5

8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	50.0	75.0	95.0	57.5	75.0	102.5	60.0	75.0	100.0
Total	220.0			235.0			235.0		

For output response evaluation, use comparators

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	57.5	75.0	95.0	65.0	75.0	102.5	67.5	75.0	100.0
Total	227.5			242.5			242.5		

For output response evaluation, use window detectors

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	57.5	75.0	95.0	65.0	75.0	102.5	67.5	75.0	100.0
Total	227.5			242.5			242.5		

For output response evaluation, use peak detectors

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5

8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	50.0	75.0	95.0	57.5	75.0	102.5	60.0	75.0	100.0
Total	220.0			235.0			235.0		

For output response evaluation, use phase detectors

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	50.0	75.0	95.0	57.5	75.0	102.5	60.0	75.0	100.0
Total	220.0			235.0			235.0		

For output response evaluation, use sample and hold circuits

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	0	0	0	0	0	0	0	0	0
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	52.5	62.5	80.0	60.0	62.5	87.5	62.5	62.5	85.0
Total	195.0			210.0			210.0		

For output response evaluation, use A/D conversion and a processor

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	0	0	0	0	0	0	0	0	0

8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	7.5	0	0	7.5	0	0	7.5	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	37.5	50.0	70.0	40.0	50.0	77.5	42.5	50.0	75.0
Total	157.5			167.5			167.5		

For output response evaluation, use A/D conversion and a magnitude accumulator

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	40.0	50.0	75.0	42.5	50.0	82.5	45.0	50.0	80.0
Total	165.0			175.0			175.0		

For coupling a circuit to a response evaluator, use existing antennas

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	10	15	5	15	15	10	20	15	15
3	5	5	15	10	5	20	15	5	25
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	47.5	57.5	65.0	62.5	57.5	82.5	77.5	57.5	95.0
Total			170.0			202.5			230.0

For coupling a circuit to a response evaluator, use analog multiplexing

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	0	0	0	0	0	0	0	0	0

8	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	35.0	55.0	72.5	42.5	55.0	85.0	50.0	55.0	87.5
Total	162.5			182.5			192.5		

For coupling a circuit to a response evaluator, use RF relays

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	17.5	27.5	30.0	20.0	27.5	35.0	22.5	27.5	35.0
Total	75.0			82.5			85.0		

For coupling a circuit to a response evaluator, use directional couplers

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	22.5	35.0	45.0	30.0	35.0	55.0	37.5	35.0	60.0
Total	102.5			120.0			132.5		

For coupling a circuit to a response evaluator, use transformers

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	0	0	0	0	0	0	0	0	0

8	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	27.5	42.5	50.0	35.0	42.5	62.5	42.5	42.5	65.0
Total	120.0			140.0			150.0		

III-13. Analog Structured DFT Techniques

For mixed signal chips, use internal scan path in the digital section

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0
Sub	42.5	42.5	60.0	50.0	42.5	62.5	52.5	42.5	65.0
Total			145.0			155.0			160.0

For mixed signal circuit boards, use boundary scan (IEEE 1149.1) in the digital section

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15

7	15	15	15	15	15	10	10	15	5
8	10	10	15	10	10	10	5	10	5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	0	0	0	0	0	0	0	0	0
14	5	5	5	5	5	7.5	5	5	10
Sub	62.5	67.5	87.5	70.0	67.5	87.5	67.5	67.5	87.5
Total	217.5			225.0			222.5		

For mixed signal circuits, use analog multiplexing techniques for partitioning

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
14	5	5	5	5	5	7.5	5	5	10
Sub	32.5	45.0	52.5	37.5	45.0	55.0	37.5	45.0	57.5
Total	130.0			137.5			140.0		

For analog circuits, use analog "scan path" techniques

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
14	5	5	5	5	5	7.5	5	5	10
Sub	32.5	42.5	52.5	40.0	42.5	57.5	42.5	42.5	62.5
Total	127.5			140.0			147.5		

For analog circuit boards, use analog test buses (IEEE 1149.4)

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	15	15	15	15	15	10	10	15	5

8	10	10	15	10	10	10	5	10	5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	0	0	0	0	0	0	0	0	0
14	5	5	5	5	5	7.5	5	5	10
Sub	52.5	67.5	87.5	60.0	67.5	87.5	57.5	67.5	87.5
Total	207.5			215.0			212.5		

III-14. Analog Ad-hoc DFT Techniques

Use physical partitioning

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	0	0	0	0	0	0	0	0	0
4	10	0	0	10	0	0	10	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	15	0	0	15	0	0	15	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	77.5	70.0	92.5	82.5	70.0	92.5	80.0	70.0	87.5
Total			240.0			245.0			237.5

Use electronic partitioning

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	0	0	0	0	0	0	0	0	0
4	10	0	0	10	0	0	10	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15

7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	15	0	0	15	0	0	15	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	77.5	70.0	92.5	82.5	70.0	92.5	80.0	70.0	87.5
Total	240.0			245.0			237.5		

Use test points for controllability, observability, and calibration

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	0	0	0	0	0	0	0	0	0
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	5	5	15	5	5	15	5	5	15
6	0	0	0	0	0	0	0	0	0
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	10	10	15	10	10	10	5	10	5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	15	0	0	15	0	0	15	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	10	10	10	10	10	15	10	10	20
Sub	70.0	65.0	97.5	77.5	65.0	102.5	77.5	65.0	102.5
Total	232.5			245.0			245.0		

Use buffering techniques for test points

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	0	0	0	0	0	0	0	0	0
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	0	0	0	0	0	0	0	0	0
14	10	10	10	10	10	15	10	10	20
Sub	52.5	45.0	67.5	57.5	45.0	77.5	62.5	45.0	87.5
Total			165.0			180.0			195.0

Ensure proper size, shape, and location of test pads

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5

8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	0	0	0	0	0	0	0	0	0
14	5	5	5	5	5	7.5	5	5	10
Sub	70.0	72.5	95.0	80.0	72.5	102.5	85.0	72.5	110.0
Total	237.5			255.0			267.5		

Use adequate ground pads

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	0	0	0	0	0	0	0	0	0
14	5	5	5	5	5	7.5	5	5	10
Sub	57.5	60.0	80.0	67.5	60.0	92.5	77.5	60.0	105.0
Total	197.5			220.0			242.5		

Ensure proper and compatible test connector design

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	10	15	5	15	15	10	20	15	15
3	5	5	15	10	5	20	15	5	25
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	10	10	10	10	10	15	10	10	20
Sub	82.5	77.5	107.5	95.0	77.5	117.5	100.0	77.5	122.5
Total			267.5			290.0			300.0

Ensure no high voltage or high frequency signals exist in the test interface

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	10	15	5	15	15	10	20	15	15
3	5	5	15	10	5	20	15	5	25
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5

8	0	0	0	0	0	0	0	0	0
9	5	10	15	10	10	20	15	10	25
10	15	0	0	15	0	0	15	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	0	0	0	0	0	0	0	0	0
14	5	5	5	5	5	7.5	5	5	10
Sub	80.0	80.0	87.5	100.0	80.0	107.5	117.5	80.0	127.5
Total	247.5			287.5			325.0		

Ensure direct test access to A/D and D/A converters

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	5	5	15	10	5	20	15	5	25
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	10	10	15	10	10	10	5	10	5
9	0	0	0	0	0	0	0	0	0
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	10	10	10	10	10	15	10	10	20
Sub	75.0	82.5	107.5	85.0	82.5	112.5	85.0	82.5	112.5
Total	265.0			280.0			280.0		

Ensure direct test access to transducers

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	0	0	0	0	0	0	0	0	0
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	10	10	15	10	10	10	5	10	5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	10	10	10	10	10	15	10	10	20
Sub	70.0	77.5	105.0	77.5	77.5	107.5	75.0	77.5	105.0
Total			252.5			262.5			257.5

Ensure direct test access to driver elements

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	0	0	0	0	0	0	0	0	0
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	0	0	0	0	0	0	0	0	0
7	15	15	15	15	15	10	10	15	5

8	10	10	15	10	10	10	5	10	5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	62.5	65.0	92.5	70.0	65.0	92.5	67.5	65.0	87.5
Total	220.0			227.5			220.0		

Break global feedback paths

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	0	0	0	0	0	0	0	0	0
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	42.5	40.0	67.5	45.0	40.0	67.5	42.5	40.0	62.5
Total	150.0			152.5			145.0		

Break long functional paths

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	0	0	0	0	0	0	0	0	0
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	42.5	40.0	67.5	45.0	40.0	67.5	42.5	40.0	62.5
Total	150.0			152.5			145.0		

Isolate complex circuits or components for separate test

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	15	15	15	15	15	10	10	15	5

8	10	10	15	10	10	10	5	10	5
9	5	10	15	10	10	20	15	10	25
10	15	0	0	15	0	0	15	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	5	5	5	5	5	7.5	5	5	10
Sub	62.5	60.0	85.0	67.5	60.0	82.5	62.5	60.0	70.0
Total			207.5			210.0			192.5

Inhibit free-running clocks

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	20	0	0	20	0	0	20	0	0
5	0	0	0	0	0	0	0	0	0
6	2.5	7.5	7.5	2.5	7.5	7.5	2.5	7.5	7.5
7	15	15	15	15	15	10	10	15	5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	0	0	0	0	0	0	0	0	0
12	5	10	15	5	10	15	5	10	15
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	5	5	5	5	5	7.5	5	5	10
Sub	77.5	72.5	77.5	87.5	72.5	82.5	90.0	72.5	82.5
Total			227.5			242.5			245.0

Avoid single points of failure

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	0	0	0	0	0	0	0	0	0
2	10	15	5	15	15	10	20	15	15
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	5	5	7.5	5	5	5	2.5	5	2.5
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	0	0	0	0	0	0	0	0	0
14	5	5	5	5	5	7.5	5	5	10
Sub	22.5	30.0	25.0	27.5	30.0	30.0	30.0	30.0	35.0
Total	77.5			87.5			95.0		

Minimize manual adjustments

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	10	15	5	15	15	10	20	15	15
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5

8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	5	0	10	5	0	10	5	0	10
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0
Sub	65.0	80.0	87.5	80.0	80.0	97.5	90.0	80.0	107.5
Total			232.5			257.5			277.5

Terminate unused inputs

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	10	15	5	15	15	10	20	15	15
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	20	0	0	20	0	0	20	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	0	0	0	0	0	0	0	0	0
Sub	80.0	75.0	92.5	92.5	75.0	100.0	100.0	75.0	102.5
Total			247.5			267.5			277.5

Avoid select-on-test components

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7	0	0	0	0	0	0	0	0	0
8	10	10	15	10	10	10	5	10	5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	0	0	0	0	0	0	0	0	0
11	5	0	10	5	0	10	5	0	10
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	5	10	15	5	10	15	5	10	5
14	0	0	0	0	0	0	0	0	0
Sub	62.5	80.0	100.0	75.0	80.0	107.5	82.5	80.0	105.0
Total			242.5			262.5			267.5

Avoid electromechanical devices

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	5	5	15	10	5	20	15	5	25
4	10	0	0	10	0	0	10	0	0
5	5	5	15	5	5	15	5	5	15
6	5	15	15	5	15	15	5	15	15
7	0	0	0	0	0	0	0	0	0

8	5	5	7.5	5	5	5	2.5	5	2.5
9	5	10	15	10	10	20	15	10	25
10	0	0	0	0	0	0	0	0	0
11	5	0	10	5	0	10	5	0	10
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	0	0	0	0	0	0	0	0	0
14	5	5	5	5	5	7.5	5	5	10
Sub	62.5	77.5	97.5	80.0	77.5	115.0	95.0	77.5	132.5
Total			237.5			272.5			305.0

Ensure UUT/ATE interface integrity and compatibility

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5
8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	5	10	15	5	10	15	5	10	15
13	5	10	15	5	10	15	5	10	5
14	10	10	10	10	10	15	10	10	20
Sub	77.5	85.0	107.5	87.5	85.0	117.5	92.5	85.0	117.5
Total			270.0			290.0			295.0

Use worst case design practices

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	5	10	2.5	7.5	10	5	10	10	7.5
2	5	7.5	2.5	7.5	7.5	5	10	7.5	7.5
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	2.5	0	5	2.5	0	5	2.5	0	5
12	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0
Sub	30.0	40.0	40.0	37.5	40.0	47.5	45.0	40.0	55.0
Total	110.0			125.0			140.0		

Ensure reasonable and accurate test specifications

	Ground			Airborne			Spaceborne		
	Devel	Manuf	Field	Devel	Manuf	Field	Devel	Manuf	Field
1	10	20	5	15	20	10	20	20	15
2	0	0	0	0	0	0	0	0	0
3	2.5	2.5	7.5	5	2.5	10	7.5	2.5	12.5
4	10	0	0	10	0	0	10	0	0
5	2.5	2.5	7.5	2.5	2.5	7.5	2.5	2.5	7.5
6	5	15	15	5	15	15	5	15	15
7	7.5	7.5	7.5	7.5	7.5	5	5	7.5	2.5

8	5	5	7.5	5	5	5	2.5	5	2.5
9	2.5	5	7.5	5	5	10	7.5	5	12.5
10	7.5	0	0	7.5	0	0	7.5	0	0
11	5	0	10	5	0	10	5	0	10
12	2.5	5	7.5	2.5	5	7.5	2.5	5	7.5
13	2.5	5	7.5	2.5	5	7.5	2.5	5	2.5
14	10	10	10	10	10	15	10	10	20
Sub	72.5	77.5	92.5	82.5	77.5	102.5	87.5	77.5	107.5
Total	242.5			262.5			272.5		

Appendix IV: Sample DFT/BIT Technique Recommendations Output

Following is a sample output of DFT/BIT technique recommendations produced by TESPAD.

```
                    DFT/BIT Recommendations for SAMPLE20

  GENERAL RECOMMENDATIONS:
  ########################

  Test Architectures
  ==================
     TA1  BIST and boundary scan (1149.1/1149.4) on digital and analog/
          mixed-signal components, with fault tolerance at the board
          level and coding at the component level **
     TA2  BIST and boundary scan (1149.1/1149.4) on digital and analog/
          mixed-signal components **
     TA3  Structured DFT and boundary scan (1149.1/1149.4) on digital
          and analog/mixed-signal components **
     TA4  Adhoc DFT on digital and analog/mixed-signal components and
          boards **

     **connected through a hierarchy of test buses (1149.1 (for digital)
       1149.4 (for analog/mixed-signal) and 1149.5/TI ASP/National Scan-
       Bridge) and maintenance controllers.

  Test Architectures selected for this project
  ============================================
     TA2-1: Unit 7
       Reasons for selecting TA2:
          Criticality of Detection is Medium, and
          Criticality of Performance is Low or Medium

     TA2-2: Unit 8
       Reasons for selecting TA2:
          Criticality of Detection is Medium, and
          Criticality of Performance is Low or Medium

     TA2-3: Unit 9
       Reasons for selecting TA2:
          Criticality of Detection is Medium, and
          Criticality of Performance is Low or Medium

     TA1-1: Unit 10
       Reasons for selecting TA1:
          Criticality of Detection is High, and
          Criticality of Performance is Low
```

```
    TA2-4: Unit 12
           Unit 13
           Unit 14
      Reasons for selecting TA2:
         Criticality of Detection is Medium, and
         Criticality of Performance is Low or Medium

    TA3-1: Unit 16
      Reasons for selecting TA3:
         Criticality of Detection is Medium, and
         Criticality of Performance is High

    TA2-5: Unit 17
           Unit 18
           Unit 19
      Reasons for selecting TA2:
         Criticality of Detection is Medium, and
         Criticality of Performance is Low or Medium

    TA4-1: Unit 20
      Reasons for selecting TA4:
         Conditions not met for Test Architectures 1, 2, or 3
         Test Architecture 4 is the default


General Design-For-Test Technique Recommendations
=================================================
     1. Physical partitioning
     2. Use of Test Pads
     3. Use of Ground Pads
     4. Power Supply Isolation
     5. Test Connector
     6. Break global feedback paths
     7. Inhibition of Free-running Clocks
     8. Manual Adjustment Minimization
     9. Termination of Unused Inputs


DETAILED RECOMMENDATIONS:
#########################

Recommended as essential to achieve testability specifications
==============================================================

Unit:  Unit 17
   Level:                Board
   App Env:              Ground Mobile
   Unit type:
   A/D:                  Digital
   Sequential:           False
   Recommended Techniques:
       1.  In-Circuit Test
       2.  IEEE Standard 1149.1
       3.  On-Board ROM
```

```
Unit:  Unit 11
   Level:                    Subsystem
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Digital
   Sequential:               False
   Recommended Techniques:
       1.  IEEE Proposed Standard 1149.5
       2.  IEEE Standard 488.1
       3.  IEEE Standard 1149.1


Unit:  Unit 13
   Level:                    Component
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Digital
   Sequential:               True
   Component Desc:           Other
   Recommended Techniques:
       1.  Microprocessor Built-In Test
       2.  Cyclic Analysis Testing System (CATS)
       3.  Simultaneous Self-Test (SST)
       4.  Level-Sensitive Scan Design On-Chip Self-Test (LOCST)
       5.  Circular Built-In Self-Test


Unit:  Unit 1
   Level:                    System
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Digital
   Sequential:               True
   Recommended Techniques:
       1.  IEEE Proposed Standard 1149.5
       2.  IEEE Standard 488.1
       3.  IEEE Standard 1149.1


Recommended to ensure testability specifications; balance versus cost
=====================================================================

Unit:  Unit 8
   Level:                    Board
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Analog
   Sequential:               False
   Recommended Techniques:
       1.  Input Stimulus: Digitized Waveform
       2.  Response Evaluator: Digital Magnitude Accumulator
       3.  Response Evaluator: Digital Processor
       4.  Input Stimulus: Oversampling-based Function Generator
       5.  Stimulus Coupling: Modulator and Mixer
```

```
Unit:  Unit 18
   Level:                    Component
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Digital
   Sequential:               False
   Component Desc:           Other
   Recommended Techniques:
       1.  Cyclic Analysis Testing System (CATS)
       2.  On-Chip ROM
       3.  Simultaneous Self-Test (SST)
       4.  Level-Sensitive Scan Design On-Chip Self-Test (LOCST)
       5.  Circular Built-In Self-Test


Unit:  Unit 9
   Level:                    Board
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Analog
   Sequential:               False
   Recommended Techniques:
       1.  Input Stimulus: Digitized Waveform
       2.  Response Evaluator: Digital Magnitude Accumulator
       3.  Response Evaluator: Digital Processor
       4.  Input Stimulus: Oversampling-based Function Generator
       5.  Stimulus Coupling: Modulator and Mixer


Unit:  Unit 5
   Level:                    Subsystem
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Analog
   Sequential:               False
   Recommended Techniques:
       1.  IEEE Proposed Standard 1149.5
       2.  IEEE Standard 488.1
       3.  IEEE Standard 1149.1


Unit:  Unit 3
   Level:                    Subsystem
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Digital
   Sequential:               True
   Recommended Techniques:
       1.  IEEE Proposed Standard 1149.5
       2.  IEEE Standard 488.1
       3.  IEEE Standard 1149.1
```

```
Unit:  Unit 19
   Level:                 Component
   App Env:               Ground Mobile
   Unit type:
   A/D:                   Digital
   Sequential:            False
   Component Desc:        Other
   Recommended Techniques:
       1.  Cyclic Analysis Testing System (CATS)
       2.  On-Chip ROM
       3.  Simultaneous Self-Test (SST)
       4.  Level-Sensitive Scan Design On-Chip Self-Test (LOCST)
       5.  Circular Built-In Self-Test


Unit:  Unit 14
   Level:                 Component
   App Env:               Ground Mobile
   Unit type:
   A/D:                   Digital
   Sequential:            False
   Component Desc:        Other
   Recommended Techniques:
       1.  Cyclic Analysis Testing System (CATS)
       2.  On-Chip ROM
       3.  Simultaneous Self-Test (SST)
       4.  Level-Sensitive Scan Design On-Chip Self-Test (LOCST)
       5.  Circular Built-In Self-Test


Unit:  Unit 2
   Level:                 Subsystem
   App Env:               Ground Mobile
   Unit type:
   A/D:                   Mixed
   Sequential:            False
   Recommended Techniques:
       1.  IEEE Proposed Standard 1149.5
       2.  IEEE Standard 488.1
       3.  IEEE Standard 1149.1


Unit:  Unit 12
   Level:                 Board
   App Env:               Ground Mobile
   Unit type:
   A/D:                   Digital
   Sequential:            False
   Recommended Techniques:
       1.  In-Circuit Test
       2.  IEEE Standard 1149.1
       3.  On-Board ROM
```

Unit: Unit 15
Level: System
App Env: Ground Mobile
Unit type:
A/D: Digital
Sequential: True
Recommended Techniques:

1. IEEE Proposed Standard 1149.5
2. IEEE Standard 488.1
3. IEEE Standard 1149.1

Unit: Unit 7
Level: Board
App Env: Ground Mobile
Unit type:
A/D: Analog
Sequential: False
Recommended Techniques:

1. Input Stimulus: Digitized Waveform
2. Response Evaluator: Digital Magnitude Accumulator
3. Response Evaluator: Digital Processor
4. Input Stimulus: Oversampling-based Function Generator
5. Stimulus Coupling: Modulator and Mixer

Unit: Unit 10
Level: Component
App Env: Ground Mobile
Unit type:
A/D: Digital
Sequential: True
Component Desc: Other
Recommended Techniques:

1. Microprocessor Built-In Test
2. Cyclic Analysis Testing System (CATS)
3. Simultaneous Self-Test (SST)
4. Level-Sensitive Scan Design On-Chip Self-Test (LOCST)
5. Circular Built-In Self-Test

1. Cyclic Codes
2. Hamming Codes
3. Arithmetic Codes
4. Checksums
5. Berger Codes

```
Unit:  Unit 20
   Level:                    Component
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Digital
   Sequential:               True
   Component Desc:           Other
   Recommended Techniques:
       1.  Provide initialization capability
       2.  Avoid redundant logic
       3.  Disable free running system clocks during test
       4.  Limit chip fanout at test points to one less than maximum
       5.  Avoid asynchronous design practices


Unit:  Unit 6
   Level:                    Subsystem
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Digital
   Sequential:               False
   Recommended Techniques:
       1.  IEEE Proposed Standard 1149.5
       2.  IEEE Standard 488.1
       3.  IEEE Standard 1149.1


Unit:  Unit 4
   Level:                    Subsystem
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Digital
   Sequential:               False
   Recommended Techniques:
       1.  IEEE Proposed Standard 1149.5
       2.  IEEE Standard 488.1
       3.  IEEE Standard 1149.1


Unit:  Unit 16
   Level:                    Component
   App Env:                  Ground Mobile
   Unit type:
   A/D:                      Digital
   Sequential:               True
   Component Desc:           Other
   Recommended Techniques:
       1.  CrossCheck
       2.  Pseudo-Exhaustive Test
       3.  LSSD
       4.  Quiescent Power Supply Current Test (IDDQ Test)
```

Appendix V: Sample Documentation Template

Following is a sample output of a documentation template produced by TESPAD.

```
units 1-3,5-20 -- Procured
################
1. Documentation of fault models
      Description of fault model assumptions used

2. Documentation of fault sampling
      Description of fault sampling procedures used

3. Documentation for operation environment requirements
      Description of operating environment factors affecting
      functional performance or test performance

4. Documentation of test procedures
      Description of all test procedures and equipment used

5. Documentation of fault isolation strategy employed
      Description of all fault isolation strategies and assumptions used


unit 4 -- COTS -- aa
################
1. Documentation for operation environment requirements
      Description of operating environment factors affecting
      functional performance or test performance

2. Documentation of test features of off-the-shelf components used
      Description of all test features, including internal test features
      and external test requirements and interfaces


If available without additional cost:
3. Documentation of fault models
      Description of fault model assumptions used

4. Documentation of fault sampling
      Description of fault sampling procedures used

5. Documentation of test procedures
      Description of all test procedures and equipment used

6. Documentation of fault isolation strategy employed
      Description of all fault isolation strategies and assumptions used
```

Appendix VI: Sample Metric Template

Following is a sample output of a metric template produced by TESPAD.

```
unit 1
################
1. FFD
    a. Test Phase:  Undefined
    b. Test Means:  Mixed
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          None
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
          Undefined
    g. Quantitative Requirements:
          FFD = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up

2. FFI
    a. Test Phase:  field
    b. Test Means:  Mixed
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          None
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
                number of failures isolated to a repairable unit in time
          FFI = --------------------------------------------------------
                            total number of failures in time
    g. Quantitative Requirements:
          FFI = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling

3. Estimated Ambiguity Group Size
    a. Test Phase:  Undefined
    b. Test Means:  Mixed
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          None
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
             Summation over failures i of (Ambiguity group size for failure i)
      E{AG} = ----------------------------------------------------------------
                             total number of failures (i)
    g. Quantitative Requirements:
          E{AG} = 0.000000
```

```
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling


unit 2
################
1. FFD
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
          Undefined
    g. Quantitative Requirements:
          FFD = 0.950045
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up

2. FFI
    a. Test Phase:  field
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
                number of failures isolated to a repairable unit in time
          FFI = ---------------------------------------------------------
                            total number of failures in time
    g. Quantitative Requirements:
          FFI = 1.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling

3. Estimated Ambiguity Group Size
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
            Summation over failures i of (Ambiguity group size for failure i)
      E{AG} = ---------------------------------------------------------------
                              total number of failures (i)
    g. Quantitative Requirements:
          E{AG} = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling
```

```
unit 3
################
1. FFD
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
          Undefined
    g. Quantitative Requirements:
          FFD = 0.200000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up

2. FFI
    a. Test Phase:  field
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
                number of failures isolated to a repairable unit in time
          FFI = ---------------------------------------------------------
                            total number of failures in time
    g. Quantitative Requirements:
          FFI = 0.020000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling

3. Estimated Ambiguity Group Size
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
             Summation over failures i of (Ambiguity group size for failure i)
      E{AG} = -----------------------------------------------------------------
                              total number of failures (i)
    g. Quantitative Requirements:
          E{AG} = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling
```

```
unit 4
################
1. FFD
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
          Undefined
    g. Quantitative Requirements:
          FFD = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up

2. FFI
    a. Test Phase:  field
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
                number of failures isolated to a repairable unit in time
          FFI = --------------------------------------------------------
                             total number of failures in time
    g. Quantitative Requirements:
          FFI = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling

3. Estimated Ambiguity Group Size
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
             Summation over failures i of (Ambiguity group size for failure i)
      E{AG} = -----------------------------------------------------------------
                                total number of failures (i)
    g. Quantitative Requirements:
          E{AG} = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling
```

```
unit 5
################
1. FFD
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
          Undefined
    g. Quantitative Requirements:
          FFD = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up


2. FFI
    a. Test Phase:  field
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
                number of failures isolated to a repairable unit in time
          FFI = --------------------------------------------------------
                            total number of failures in time
    g. Quantitative Requirements:
          FFI = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling

3. Estimated Ambiguity Group Size
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
             Summation over failures i of (Ambiguity group size for failure i)
      E{AG} = -----------------------------------------------------------------
                              total number of failures (i)
    g. Quantitative Requirements:
          E{AG} = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling
```

```
unit 6
################
1. FFD
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
         Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
         Single stuck-at
    f. Quantitative Definition of the Metric:
         Undefined
    g. Quantitative Requirements:
         FFD = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
         Simulation, hot mock-up

2. FFI
    a. Test Phase:  field
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
         Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
         Single stuck-at
    f. Quantitative Definition of the Metric:
               number of failures isolated to a repairable unit in time
         FFI = ---------------------------------------------------------
                           total number of failures in time
    g. Quantitative Requirements:
         FFI = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
         Simulation, hot mock-up, topological dependency modeling

3. Estimated Ambiguity Group Size
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
         Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
         Single stuck-at
    f. Quantitative Definition of the Metric:
            Summation over failures i of (Ambiguity group size for failure i)
      E{AG} = ------------------------------------------------------------------
                             total number of failures (i)
    g. Quantitative Requirements:
         E{AG} = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
         Simulation, hot mock-up, topological dependency modeling
```

```
unit 7
################
1. FFD
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Bit-flip
    f. Quantitative Definition of the Metric:
          Undefined
    g. Quantitative Requirements:
          FFD = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up

2. FFI
    a. Test Phase:  field
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Bit-flip
    f. Quantitative Definition of the Metric:
                number of failures isolated to a repairable unit in time
          FFI = --------------------------------------------------------
                            total number of failures in time
    g. Quantitative Requirements:
          FFI = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling

3. Estimated Ambiguity Group Size
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Bit-flip
    f. Quantitative Definition of the Metric:
             Summation over failures i of (Ambiguity group size for failure i)
      E{AG} = -------------------------------------------------------------------
                              total number of failures (i)
    g. Quantitative Requirements:
          E{AG} = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling
```

```
unit 8
################
1. FFD
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
         Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
         Bit-flip
    f. Quantitative Definition of the Metric:
         Undefined
    g. Quantitative Requirements:
         FFD = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
         Simulation, hot mock-up

2. FFI
    a. Test Phase:  field
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
         Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
         Bit-flip
    f. Quantitative Definition of the Metric:
               number of failures isolated to a repairable unit in time
       FFI = ---------------------------------------------------------
                            total number of failures in time
    g. Quantitative Requirements:
         FFI = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
         Simulation, hot mock-up, topological dependency modeling


3. Estimated Ambiguity Group Size
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
         Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
         Bit-flip
    f. Quantitative Definition of the Metric:
              Summation over failures i of (Ambiguity group size for failure i)
     E{AG} = -----------------------------------------------------------------
                               total number of failures (i)
    g. Quantitative Requirements:
         E{AG} = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
         Simulation, hot mock-up, topological dependency modeling
```

```
unit 9
################
1. FFD
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
         Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
         Bit-flip
    f. Quantitative Definition of the Metric:
         Undefined
    g. Quantitative Requirements:
         FFD = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
         Simulation, hot mock-up

2. FFI
    a. Test Phase:  field
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
         Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
         Bit-flip
    f. Quantitative Definition of the Metric:
               number of failures isolated to a repairable unit in time
         FFI = --------------------------------------------------------
                           total number of failures in time
    g. Quantitative Requirements:
         FFI = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
         Simulation, hot mock-up, topological dependency modeling

3. Estimated Ambiguity Group Size
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
         Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
         Bit-flip
    f. Quantitative Definition of the Metric:
            Summation over failures i of (Ambiguity group size for failure i)
     E{AG} = -----------------------------------------------------------------
                             total number of failures (i)
    g. Quantitative Requirements:
         E{AG} = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
         Simulation, hot mock-up, topological dependency modeling
```

```
unit 10
################
1. FFD
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
          Undefined
    g. Quantitative Requirements:
          FFD = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up

2. FFI
    a. Test Phase:  field
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
                number of failures isolated to a repairable unit in time
          FFI = ---------------------------------------------------------
                            total number of failures in time
    g. Quantitative Requirements:
          FFI = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling

3. Estimated Ambiguity Group Size
    a. Test Phase:  Undefined
    b. Test Means:  ATE
    c. Test Mode:   Off-line
    d. Degree of Allowable External Support:
          Operator may be 'in the loop' to help achieve requirement
    e. Fault Model Assumptions:
          Single stuck-at
    f. Quantitative Definition of the Metric:
             Summation over failures i of (Ambiguity group size for failure i)
      E{AG} = -----------------------------------------------------------------
                              total number of failures (i)
    g. Quantitative Requirements:
          E{AG} = 0.000000
    h. Allowable Requirements Compliance Tracking Methodology
          Simulation, hot mock-up, topological dependency modeling
```

```
unit 11
################
1. FFD
     a. Test Phase:  Undefined
     b. Test Means:  ATE
     c. Test Mode:   Off-line
     d. Degree of Allowable External Support:
            Operator may be 'in the loop' to help achieve requirement
     e. Fault Model Assumptions:
            Single stuck-at
     f. Quantitative Definition of the Metric:
            Undefined
     g. Quantitative Requirements:
            FFD = 0.000000
     h. Allowable Requirements Compliance Tracking Methodology
            Simulation, hot mock-up

2. FFI
     a. Test Phase:  field
     b. Test Means:  ATE
     c. Test Mode:   Off-line
     d. Degree of Allowable External Support:
            Operator may be 'in the loop' to help achieve requirement
     e. Fault Model Assumptions:
            Single stuck-at
     f. Quantitative Definition of the Metric:
                    number of failures isolated to a repairable unit in time
            FFI = --------------------------------------------------------
                                    total number of failures in time
     g. Quantitative Requirements:
            FFI = 0.000000
     h. Allowable Requirements Compliance Tracking Methodology
            Simulation, hot mock-up, topological dependency modeling

3. Estimated Ambiguity Group Size
     a. Test Phase:  Undefined
     b. Test Means:  ATE
     c. Test Mode:   Off-line
     d. Degree of Allowable External Support:
            Operator may be 'in the loop' to help achieve requirement
     e. Fault Model Assumptions:
            Single stuck-at
     f. Quantitative Definition of the Metric:
                Summation over failures i of (Ambiguity group size for failure i)
        E{AG} = -----------------------------------------------------------------
                                    total number of failures (i)
     g. Quantitative Requirements:
            E{AG} = 0.000000
     h. Allowable Requirements Compliance Tracking Methodology
            Simulation, hot mock-up, topological dependency modeling
```

Appendix VII: Sample DFT/BIT Technique Template

Following is a sample output of a DFT/BIT technique template produced for BILBO by TESPAD.

1. Technique Name

   Built-In Logic Block Observation (BILBO)

2. Definition

   Built-In Logic Block Observation (BILBO) is a design technique for general synchronous sequential systems which combines the functions of the PRPG/MISR and scan design techniques. A BILBO design allows separate off-line tests to be performed on blocks of combinational logic and memory elements using multifunctional BILBO registers. All system registers are replaced with BILBO registers, which are strung together to form a scan chain. Each BILBO register can perform four functions: latch (normal operation), shift register (scan operation), multiple-input shift register (for pseudorandom pattern generation or signature compaction), and reset (set all elements to zero). Using these four functions, a circuit can be fully tested with either pseudorandom or deterministic test patterns.

3. Description

   In order to implement this technique, each system register should be replaced with a BILBO register. The BILBO register uses extra control logic to configure it into one of its four modes. When in the multiple-input shift register (MISR) mode, the BILBO can be used either as a pseudorandom pattern generator (PRPG) or a signature analyzer (SA), depending on its parallel inputs. When a circuit provides inputs to the MISR, it acts as a SA, but if the inputs remain constant, the MISR acts as a PRPG. Many BILBO systems are modular and bus-oriented, so by disabling bus drivers and using pull-up or pull-down circuitry, the BILBO inputs can be held constant during the test mode. If such design is not feasible, then an extra control line can be added to the BILBO registers to effect an input disabling state.

   A typical test session occurs in the following steps:

   1. An "Initiate BIT" signal is generated. The pass/fail flag is reset to "pass."
   2. The BILBOs are configured as shift registers, and an initial seed pattern is shifted into them.
   3. The BILBOs are configured as MISRs for use as either PRPGs or SAs.
   4. The registers are clocked, and pseudorandom testing is performed.
   5. The BILBOs are configured as shift registers and the contents are shifted out.
   6. The circuit signatures are compared to the expected values and a pass/fail diagnosis is made.

   A modification to the basic BILBO structure can be used if it becomes advantageous to use a BILBO register as both a PRPG and SA simultaneously. The L3-BILBO uses two independent slave latches for each master latch, each of which has separate routing connections. This simplifies the test procedure, but increases area overhead, requires additional clocks, and decreases testing speed by half.

4. Similar Techniques
   Scan Design
   Test with Pseudorandom Pattern Generator/Multiple-Input Shift Register

5. Generic Design Procedure
   1. Implement a scan-chain architecture that includes all registers.

   2. Replace all system registers with BILBO registers.

   3. Provide external access to the BILBO control lines so that the test mode can be controlled.

6. Area Overhead
   The logic gate overhead for implementing a BILBO design tends to be high, due to the large amount of extra logic required to implement a BILBO register, and the fact that many such registers are needed. Some overhead can be saved, however, by observing that not all system registers may need to have all of the BILBO functions available.

7. I/O Overhead
   Two additional primary inputs/outputs are required for control of the BILBO functions. Two more are necessary to implement a scan chain for state initialization and response observation. Another pin is necessary if an input disable state is needed.

8. Power Overhead
   Since the gate overhead is high, and most of the BIT circuitry is in operation during the normal mode, the power consumption is also high.

9. Computational Expense of Design Translation
   Replacement of the system registers with BILBO registers is a straightforward task. The more difficult procedure is deciding upon a scan chain flip-flop placement and routing order. The number of test patterns needed to obtain the desired fault coverage, and the expected signature of each logic block after this test sequence must be calculated. Also, implementing a controller to schedule the tests may require significant effort.

   Tools: none

10. Available COTS Parts

11. Performance Degradation
    In a BILBO design there will be significant performance degradation in the form of throughput reduction as a result of additional gate delays introduced by replacing all system registers with BILBOs.

12. Impact on Run Time
    When compared to I/O tests on the random sequential logic, a BILBO design requires fewer test vectors to obtain a specified fault coverage since embedded registers have more direct access to much of the internal logic. Since the pseudorandom tests can be performed on all blocks in parallel, no additional test time is required. Tests can be run at the normal operating speed of the circuit.

13. Estimated Fault Detection Coverage
    Excellent fault coverage, provided that all registers have been replaced with BILBOs. A near 100% fault coverage is expected, since all registers are tested through the serial chain, and all combinational blocks are tested with pseudorandom test patterns, and can also be tested deterministically if necessary. A BILBO system is capable of detecting some delay faults since the generated test patterns are applied to the circuit under test at normal system clock speeds, and can sensitize the worst-case register-to-register paths that are exercised in normal circuit operation.

14. Estimated Fault Isolation Coverage

15. Impact on Availability, Reliability, and Maintainability
    The reliability of a chip using this technique decreases due to the large additional area required, but the maintainability and availability at higher levels of hierarchy are impacted positively because of the high fault coverage provided.

16. Applicability to Circuit Structures
    sequential

    The BILBO testing methodology is applicable to synchronous sequential logic. Strictly combinational circuits can be tested only if additional registers are included in the design.

17. Applicability to Test Modes
    off-line

18. Compatibility with other DFT/BIT Techniques
    Complementary techniques: Scan Design
    BIST for highly regular structures such as RAMs and PLAs

19. Compatibility with JTAG
    BILBO design can be made compatible with IEEE Standard 1149.1. In particular, the internal scan registers can be configured as test data registers for controllability and observability during test mode.

20. ATE Accessibility
Two lines are required at the ATE interface for control of the BILBO registers. Two more are required for the scan chain I/Os. The registers can be directly controlled and observed via the scan chain.

21. Life Cycle Use
manufacturing field

Since BILBOs must be incorporated as part of the original design of the CUT, they are useful at any part of the system life cycle. However, this method is less useful during the prototyping phase since it is very difficult to diagnose circuit failures using a compressed circuit signature.

22. Effectiveness and Priority

23. Case History 1

24. Case History 1 Description

25. Case History 2

26. Case History 2 Description

27. Case History 3

28. Case History 3 Description

29. Reference 1
M. Abramovici, M.A. Breuer, and A.D. Friedman, Digital Systems Testing and Testable Design, Computer Science Press, New York, NY: 1990.

30. Reference 2
B. Konemann, J. Mucha, and G. Zwiehoff, "Built-In Logic Block Observation Technique," Digest of Papers 1979 Test Conf., October 1979, pp. 37-41.

31. Reference 3
B. Konemann, J. Mucha, and G. Zwiehoff, "Built-In Test for Complex Digital Integrated Circuits," Journal Solid State Circuits, Vol. SC-15, No. 3, June 1980, pp. 315-318.

☆U.S. GOVERNMENT PRINTING OFFICE: 1998-610-130-61136

MISSION
OF
ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.