2

ARMSTRONG LABORATORY

RAMCAD DESIGN METHODOLOGY

Charles T. Kitzmiller
Michael W. Anderson

BOEING COMPUTER SERVICES
RESEARCH AND TECHNOLOGY
P.O. BOX 24346, M.S. 7L-64
SEATTLE, WA 98124-0346

Matthew C. Tracy, II

HUMAN RESOURCES DIRECTORATE
LOGISTICS RESEARCH DIVISION
2698 G STREET
WRIGHT-PATTERSON AFB, OH 45433-7604

MAY 1993

INTERIM TECHNICAL PAPER FOR PERIOD APRIL 1988 - DECEMBER 1992

Approved for public release; distribution is unlimited.

AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation, or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this paper and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

Matt C Tracy II

MATTHEW C. TRACY, II
Contract Monitor

BERTRAM W. CREAM, Chief
Logistics Research Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
May 1993

3. REPORT TYPE AND DATES COVERED
Interim - Apr 1988 - Dec 1992

4. TITLE AND SUBTITLE
RAMCAD Design Methodology

5. FUNDING NUMBERS
C - F33615-87-C-0001
PE - 63106F
PR - 2940
TA - 01
WU - 04

6. AUTHOR(S)
Charles T. Kitzmiller
Matthew C. Tracy, II
Michael W. Anderson

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

Boeing Computer Services
Research and Technology
P.O. 24346, M.S. 7L-64
Seattle, WA 98124-0346

Armstrong Laboratory
Human Resources Directorate
Logistics Research Division
2698 G Street
WPAFB, OH 45433-7604

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Armstrong Laboratory
Human Resources Directorate
Logistics Research Division
2698 G Street
Wright-Patterson AFB, OH 45433-7604

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AL/HR-TP-1993-0017

11. SUPPLEMENTARY NOTES
Armstrong Laboratory Technical Monitor: Matthew C. Tracy, AL/HRGA, (513) 255-8399

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The goal of the Reliability, Availability, and Maintainability in Computer-Aided Design (RAMCAD) research effort is to define a design environment that fully supports reliability, maintainability, and supportability issues through the use of computer-aided design/computer-aided engineering workstations. As part of this effort, the current design methodology was analyzed and a new variation of the methodology was proposed. This report describes the proposed methodology as well as the computer support requirements, measures of effectiveness, and design techniques that are required to implement the methodology.

14. SUBJECT TERMS
computer-aided design
computer-aided engineering
design
design process
maintainability
methodology
optimization
RAMCAD
reliability

15. NUMBER OF PAGES
131

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	SAR

CONTENTS

FIGURES

Figure Page

TABLES

Table Page

PREFACE

To achieve the goals of Department of Defense (DoD) programs such as Reliability and Maintainability 2000, the Air Force needs to greatly improve weapon system designs. To help meet these DoD program goals and improve weapon system designs, the Armstrong Laboratory, Logistics Research Division (AL/HRG) embarked on the Reliability, Availability, and Maintainability in Computer-Aided Design (RAMCAD) effort. The goal of this effort is to create a design environment that fully supports analyzing designs for and responding to reliability, maintainability, and supportability (RM&S) concerns through the use of computer-aided design/computer-aided engineering (CAD/CAE) workstations and software.

In September 1987, AL/HRG awarded contract F33615-87-C-0001 to Boeing Computer Services (BCS) to perform long-term research associated with the RAMCAD effort. The goals of the contract included developing a methodology for local design optimization and creating proof-of-concept software tools that implement the methodology on a CAD/CAE workstation. BCS focused its efforts on a methodology and tools to support locally optimizing an electronic design with respect to reliability, maintainability, and testability as well as functionality, performance, cost, power, and area.

This report describes the methodology that was created under the BCS RAMCAD contract. The methodology helps design engineers integrate responses to RM&S concerns and requirements into the design of a system more thoroughly and at an earlier point in the design cycle. This is accomplished by including tools in the CAD/CAE environment that help design engineers locally optimize a design with respect to one or more requirements. This report includes discussions on problem areas in the current design environment, the proposed methodology, a variety of RM&S metrics, and various techniques for addressing RM&S concerns during design.

The authors would like to thank the people who contributed to the data in this report and to the creation of the proposed methodology. In particular, thanks go Alex Bobotek and the others at BCS as well as Chuck Yount, Dr. Dan Siewiorek, and Dr. Bill Birmingham from Carnegie-Mellon University.

I. INTRODUCTION

This report summarizes part of the results of the Armstrong Laboratory, Logistics Research Division (AL/HRG), Reliability, Availability, and Maintainability in Computer-Aided Design (RAMCAD) Software Development Program (contract F33615-87-C-0001) conducted by Boeing Computer Services (BCS). The overall purpose of the RAMCAD research effort is to develop methods and tools to enable the design of improved military systems. AL/HRG is achieving this through better automation of existing design knowledge, improved computer-aided design/computer-aided engineering software capabilities, and optimization and invention of design techniques and methodologies. The goal of the research performed by BCS and described in this report was to define an overall approach to assisting engineers[1] in optimizing the design of electronic systems for various design parameters including performance, cost, schedule, reliability, and maintainability.

This report describes the overall design methodology BCS developed as part of the RAMCAD program. The goal of the methodology is to provide an improved approach to electronic system design that yields higher quality, less expensive designs. The approach proposes new design tools that enable a design engineer to locally optimize an electronic design with respect to performance, cost, reliability, maintainability, and other requirements. Once the necessary tools are created, this methodology can be applied to all phases of the design of electronic systems. Most aspects of the approach are general and can be adapted to other areas of design as well.

Scope

The major objective of this research was to define the data, models, analyses, and methods needed to enhance the direct support for the design of electronic hardware. Although there is a need to improve the methods and techniques used to develop embedded software and firmware associated with electronic hardware systems and those used to produce the maintenance and logistics support information needed to field designed system, this research effort was limited to the design of hardware. Within this domain, the research extended beyond the recognized

[1] To avoid confusion, five main terms will be used to define people who work on designs. The first term, system engineer, denotes individuals working on system-level problems that include such tasks as apportioning and allocating resources and requirements to subsystems. The second term, detailed design engineer, refers to those who design hardware implementations that perform the functions required of one section of a subsystem (e.g., one printed circuit board) as determined by system engineers. The third term, design engineer, includes both system engineers and detailed designers. The fourth term, specialty engineer, refers to those who support the design process by performing analyses and making recommendations in all other areas of the design (e.g., reliability, maintainability, and manufacturing). The last term, engineer, includes everyone in the other four categories.

problems of data sharing and modeiing and included issues associated with optimizing electronic designs.

In particular, this research focused on developing methods and tools to assist the design engineer in performing trade-off analyses to apportion and allocate design resources (e.g., area, power, and costs) and requirements from the system level, through each subsystem level, to the detailed design level. In addition, an attempt was made to identify the tools and methods the design engineer needs to assess how well a proposed design meets its requirements and to ensure it has not exceeded its allocated resources. Throughout this report the term "requirements" is used in a broad sense to include both objectives that must be met (hard requirements or constraints) and those that are desirable but negotiable (soft requirements or goals).

Approach

The methodology described in this report was developed in response to specific needs identified by the members of the Boeing design community participating in this research effort (Kitzmiuer & Anderson, 1991). The goal in addressing these needs was to define a process that embraces and enforces good design practices while providing additional opportunities for design optimization. To accomplish this goal, the methodology development was based more on the iterative application of system engineering methods than on the use of formal optimization techniques.

Two different, but related, perspectives of the design process were used to develop the proposed methodology. One was to investigate the design and analysis needs associated with assisting the system and detailed design engineers in performing their tasks. The second was to investigate the needs of the supporting specialty engineers (e.g., reliability and testability engineers). The difference in these approaches is that the first focuses on supporting design engineers, while the second emphasizes supporting the activities of various specialty engineers.

Information from two basic sources was used to develop the methodology. The first was a series of working sessions with more than 20 senior engineers working for Boeing Military Aircraft, Boeing Aerospace and Electronics, Boeing Commercial Airplanes, Boeing Advanced Systems, and BCS. The engineers participating in the sessions included specialists in systems engineering, electronic subsystem design, digital and analog circuit design, reliability, testability, electronics packaging, manufacturing technology, and integrated logistics support. The second source was various technical reports and papers, many of which are listed in the reference and bibliography sections at the end of this report.

Report Organization

The remainder of this report is organized into five sections and two appendices.

a. Section II provides an overview of the proposed design methodology. In this section several problems associated with applying traditional optimization methods to the design of electronic systems are identified and an alternative approach to aiding engineers in developing a design that is locally optimized with respect to multiple criteria is presented.

b. Section III describes the optimization methodology in detail and indicates how the design cycle should be changed to incorporate the methodology.

c. Section IV identifies several problems associated with the current design environment and the enhancements in technology and computer support needed by design engineers, including specialized support for a variety of reliability, maintainability, and supportability (RM&S) tasks and analyses.

d. Section V describes the measures of effectiveness and design metrics that should be used with the proposed methodology.

e. Section VI describes several design strategies based on the proposed methodology.

f. Appendix A provides the mathematical algorithms used to compute some of the design metrics discussed in Section V.

g. Appendix B provides a partial listing of the design heuristics, rules, and guidelines developed during the research. The listing is an example of the type of information needed to support the proposed methodology.

II. OVERVIEW

Studies of the current electronic system design process conducted as part of this research revealed several major problems which seriously impede the ability of a design team to develop a truly optimal design. One problem is a lack of timely, accurate estimates of design attributes (e.g., cost, performance, reliability, and maintainability) during many of the design tasks. Often the estimates that can be obtained are too imprecise to clearly establish which of several competing design options is superior. Major causes of this problem are the dependence of most estimation techniques on design data that are not available until late in the design process and the lack of adequate methods to estimate many of the key factors that determine the quality of a des'gn (e.g., intermittent fault rates and connector reliability). When data are available at the required level of detail, constrained engineering budgets, the unavailability of specialty engineers to perform the analyses when needed, and the time-consuming nature of many of the analyses all contribute to the lack of timely estimates.

A second problem is a lack of effective interaction between design disciplines. Large design teams usually include specialty engineers from disciplines such as functional design, physical design, manufacturing, design assurance, and logistics support. In the large military system development programs observed as part of this research, interaction among these groups was hampered by the number of specialty engineers that need to be involved ir any single design task (often 40 or more in the design of a typical line-replaceable module [LRM]); by differences in their vocabulary, information requirements, decision processes, criteria, etc.; and by the constrained flow of design information (due to the time and resources needed to prepare and document the information being exchanged). Partly due to this problem, logistics support and other design specialties have historically not had the impact on the design process that is needed to minimize the life-cycle cost (LCC) of the design.

A third problem is a lack of computer aids to assist in capturing, managing, and presenting the information associated with a design. Design analyses help identify the strengths and weaknesses of designs. Presenting the results of these analyses to the design engineers in ways that highlight these strengths and weaknesses is as important as ensuring the data are available in a timely manner.

A fourth problem is the length of time required to design and field many military and some commercial systems. Because military systems often take a decade or more to field, several issues must be contended with that are not applicable to systems requiring less time to field. First, the length of the design and development process complicates the prediction and allocation process and increases the uncertainty associated with economic and technological estimates made by the design team. If the design team is to ensure that the system is not obsolete before it is fielded and that it can be cost-effectively operated throughout its design life, the team must accurately predict how technology will evolve during the period needed to design and field the system (e.g., 10 years). The team must assume, for example, that certain technologies which are not cost-effective today will be cost-effective when the system will be produced (i.e., 10 years in this example). To ensure that the system operates cost-effectively over its design-life (e.g., 20 years), the team must design for a 30-year period (i.e., the 10 years required to field in addition and the 20-year period of performance). Clearly, many of the factors the design team must take into account are likely to change dramatically during the design and development period. In all likelihood, many of the design team's assumptions and predictions concerning mission, support infrastructure (e.g., number and location of military bases and maintenance technician skills), and technology will not be entirely accurate. Consequently, military systems often contain obsolete subsystems that do not fully satisfy the needs that exist when they are fielded and for which spare parts can be difficult and expensive to obtain.

A second issue resulting from a long development cycle is that design engineers creating military equipment tend to participate on fewer design cycles than design engineers creating equipment with a shorter development cycle and receive little or no feedback on the results of their efforts. This causes the design engineers creating military equipment to be much less experienced and capable than their counterparts.

A third issue is the procurement process employed by the Department of Defense (DoD). This process often requires different organizations to perform research, design, and development work on the system and its subsystems while receiving oversight from a program office. This issue, combined with the first issue, ensures that there is a large turnover in the people who make decisions about the program. During this time, the design rationale and lessons learned acquired by one person are often lost and cannot propagate through the entire process.

As a result, the greatest contributions to design optimization can be made by reducing the time required to design and field a system; improving estimation models and techniques (particularly those that engineers can employ to support decisions made early in the tasks they perform); providing less costly and easier access to the available design information and analyses; assisting design teams in conducting trade studies; and assisting in the management and presentation of the information developed during the design process. Doing so addresses each of the noted problems to some degree. It directly addresses the first, third, and fourth problems by presenting information to the design engineers in a manner that facilitates timely decisions. Increasing the speed and reducing the cost of information acquisition and analysis, and facilitating its flow between engineers will help design disciplines to interact more efficiently, thereby addressing the second and fourth problems. Seeking to reduce the design time and helping design engineers to acquire and interpret information when needed was a major focus of the design optimization research.

Optimization Barriers

The design of an electronic assembly (e.g., a multiple circuit card line-replaceable unit [LRU]) for a military application typically involves consideration of many competing design requirements and several dozen evaluation metrics, and requires expertise in circuit design, electronic packaging, manufacturing, design assurance, and logistics support. Because aspects of the design problem are amenable to formal optimization techniques (e.g., linear programming techniques and nonlinear equation solving techniques), it is tempting to view the design of such a system as a problem which can be addressed solely by the use of formal optimization techniques. However, there are several properties of the circuit design problem, and the majority of design problems in general, that preclude a dependence upon these techniques. Formal optimization approaches require an evaluation function for computing a figure of merit for a candidate design and assume either that the design parameters are real variables in a continuous space, or that the possible values for a variable have been enumerated or can be generated by an algorithm. In general, neither of these requirements is met for real design problems. The design space is sparsely populated and the relationship of many of the design parameters to the quality of the end product or to other design and manufacturing factors cannot be readily expressed as mathematical or probabilistic models. In many instances, the relation of these factors to the quality of a product can only be expressed as general "rules of thumb," heuristics, or qualitative design rules and constraints, such as those listed in Appendix B.

Computing a meaningful figure of merit is often impractical. One problem is that the design space associated with electronic systems is not continuous. Electronic designs are more suitably characterized by discrete variables, such as whether a particular function is implemented, whether a function is implemented as a set of discrete circuit elements or as an application-specific integrated circuit (ASIC), or whether one design is more susceptible to intermittent faults than others. Such features are not appropriately represented as continuous parameters. In addition, the relation of these alternatives to other attributes of a design, such as the manufacturing and engineering cost, cannot be accurately predicted in many instances. As a consequence, the design space is less a coordinate space than a collection of discrete design alternatives with no effective means of representing all the properties and relations of interest as continuous parameters. Approaches to discrete optimization require that the optimization routine be able to generate and evaluate candidate designs. However, the heart of the circuit design problem is the interaction among the circuit elements and, at present, it is impossible to devise an algorithm that can analyze this interaction and exhaustively generate candidate designs.

For example, if a design problem requires a signal processing function capable of communicating with other system functions, a design engineer can conceive of alternative physical designs that provide this functionality. The signal processing and communication logic of the required function could be implemented via a general purpose computer and a modem or as an ASIC. A power supply could be incorporated into the physical implementation of the function or power could be provided by an external source. The design engineer could consider several alternative processors, modems, and power supplies individually and in combination with other alternatives. Usually a design engineer repeats this process at progressively finer levels of detail until the physical design is complete. However, the process is only partially amenable with formal optimization techniques. Since there is not, at present, a general method of generating design alternatives for this problem, the intermediate design points have no meaning and there is no sense of continuity or slope between the design alternatives. The alternatives are, to a large degree, discrete and isolated from one another. In other words, solving such problems is similar to solving complex nonlinear optimization problems requiring integer solutions. However, each variable in the problem is defined by a set of integers that varies based on the value of the other variables in the problem. Defining these relationships is a daunting task for an expert and probably impossible for a novice.

Even if the design engineer could define a satisfactory evaluation function relating the design requirements, often there is insufficient information to determine the value of many of the variables (e.g., cost, weight, and reliability) that are necessarily key factors in the evaluation function. Estimates of the values of these variables early in the design process are so imprecise that design engineers are reluctant, or more likely unwilling, to base choices among design approaches on them. Moreover, creating a design is usually a problem in meeting targets—the cost, power, speed, functionality, reliability, maintainability, etc. must each meet or exceed some requirement. Design engineers find it difficult to choose among concrete choices that differ only in the degree to which they meet or exceed various design requirements unless one choice is superior in all areas. During the RAMCAD research, design engineers often expressed the opinion that they would be unlikely to surrender their decision-making authority to an evaluation function.

In addition, other aspects of the problem limit the applicability of algorithmic or formal approaches to optimization. Among them is a need to take into account variations in the cost of acquiring the data upon which the evaluation function is based and the uncertainty, quality, and unavailability of this data. For example, estimates of the reliability of a component differ in their availability, quality, and certainty depending upon component technology and other factors. Although it cannot be easily quantified, the availability and quality of reliability data for transistor-

transistor logic (TTL) components are generally considered to be better than that for gallium arsenide (GaAs) components, a less mature technology.

Proposed Approach

The proposed approach is based on the premise that the best approach to developing a truly optimal design (i.e., one that is optimal when fielded, not just as a paper design) is to identify and focus on the design parameters and aspects of the design that can be controlled and have the greatest impact on the quality of the product (i.e., the "design drivers"). For this reason, the approach relies on an exploration of design alternatives to achieve optimality; an exploration guided by heuristics and a compliance with good design rules and practices, and the results of studies to determine the "design drivers" and locally optimal solutions. Instead of a single figure of merit combining all design parameters, it is proposed that an integrated product team (IPT) composed of engineers from each major discipline establish separate requirements for each criterion defined in Section V and that, for the most part, the design process is carried out at all levels by design engineers who specialize in functional and physical design. In this approach, the role of the specialty engineer is to support the design engineer in apportioning and allocating design resources and requirements, in identifying and evaluating design alternatives, and in performing trade studies. In addition to experts in functional and physical design, the IPT should include specialists in manufacturing, reliability, maintainability, producibility, etc. Optimization is the product of the collective expertise of the IPT and the trade studies it conducts; the experience of the IPT should always take precedence over the results of any formal optimization attempts.

A key element of the proposed approach is the use of brainstorming and other consensus-building techniques early in the design process to identify the design strategies and alternatives the IPT will explore. The use of such techniques increases the likelihood that the best candidate designs are considered since they provide a means of ensuring that all members of the IPT have a voice in the design process. In addition to generating new ideas to solve the current problem, brainstorming sessions often produce ideas that form the basis for research programs and future design techniques and options.

In addition to brainstorming and consensus-building sessions, it is proposed that a significant portion of the time devoted to design be dedicated to both formal and informal design reviews. Where possible and appropriate, the design reviews should include specialty engineers that are not part of the IPT. The reviews should be structured as group discussions, include discussions of abandoned alternatives, and have two goals. One goal should be for the design engineers to receive advice about how to best address the concerns of the specialty engineers (the advice should include suggestions of new techniques shown to yield significant improvements in the specialty engineers' areas of interest, as well as warnings about approaches found to be detrimental to these areas of interest). To support this role, the specialty engineers should continually evaluate the performance of fielded designs with respect to their area of interest, analyze the problems encountered, and feed the results of their analyses back to the design engineers. The progress of the design should determine the frequency of these sessions. They should be held frequently enough to allow all participants an opportunity to comfortably review the decisions of the previous session and to preclude extensive design work from being completed based on decisions not yet reviewed.

The second goal should be to improve the expertise of the participating specialty engineers. Specialty engineers should be given opportunities to identify situations where functionally attractive designs must be abandoned because of poor performance in their area of specialization. Such situations indicate potentially fruitful areas of investigation for the IPT and design methodology researchers. The IPT as a whole should attempt to determine and catalog the source of the weakness of each design, and to search for solutions that preserve the desired functionality of each design while yielding improved performance in other areas.

There are several opportunities to provide automated support for this approach. The most fundamental one is to support the capture, comparison, and presentation of design information and data. Tools could assist the IPT in recording and justifying design choices, and in presenting these decisions and the supporting rationale to others involved in the review sessions (e.g., brainstorming and consensus-building tools, and design rationale capture tools). Tools could also assist the IPT in conducting and documenting trade studies, in collecting the data on which these studies are based, and in making and documenting these evaluations (e.g., the System for the Interactive Design and Computer Analysis of Reliability [SIDECAR] program created under this effort and described in Section V). Although the initial deployment of such tools would, in all likelihood, only provide the design engineer access to the results of completed studies performed by specialty engineers, the long term goal should be to automate, to the greatest extent possible, the analyses performed by the specialty engineers so that the design engineer can perform many of these trade studies independently. This, however, will require significant advances in knowledge-engineering methods and knowledge-based systems technology.

Finally, the use of formal optimization techniques could be automated to a limited degree as the design becomes more detailed—provided the expertise of the IPT takes precedence over the results of the optimization attempt. These techniques become more feasible because the design alternatives become more concrete and the similarities to existing designs become more exact as the design becomes more detailed. As a result, design parameter estimates based on existing designs become increasingly accurate and the evaluations of design alternatives more reliable. Also, as the design problem becomes more circumscribed, the design engineer is more likely to be able to provide an evaluation function and accept its results with confidence. At this point, support for more automated approaches to design optimization becomes possible. These approaches include performing sensitivity analyses to identify areas of the design with the greatest impact on the figure of merit and evaluating each element of a known set of equivalent functions to determine which yields the best figure of merit in the case at hand. In some instances, at the most detailed design level it may be possible to develop a parameterized design for common functions (i.e., a parametric function relating all design parameters) and apply formal optimization techniques (e.g., annealing) to optimize a selected aspect of the design within the boundaries of the parameterized space.

III. PROPOSED METHODOLOGY

This section describes an overall approach to the design of an electronic subsystem. Although the following discussion focuses on the design of an LRU or functional group of LRMs for an avionics application, the methodology is applicable to the design of electronic subsystems for other applications.

The proposed methodology addresses some of the deficiencies found in the current design process described in the paper, "Electronic Design Process" (Kitzmiller and Anderson, 1991). The basic approach is for an IPT to develop a detailed functional description of the object to be designed and to use this description as the basis for identifying and evaluating alternative physical designs (see Figure 1). At each level in a design hierarchy, the IPT proposes, evaluates, and revises, where necessary, design concepts, alternatives, and requirements in an effort to define the "design drivers" and to develop the requirements, concepts, and alternatives at a lower, more detailed level. The IPT employs separate functional and physical design hierarchies to provide a framework in which it can organize and manage design data, and to facilitate the evaluation of design alternatives.

The IPT needs such a framework to encourage an initial focus on the functional design while providing an integrated framework in which both functional and physical design can be performed. The IPT also needs this framework to support changes in the focus of the design tasks as the design progresses from the conceptual to the detailed design phase and from a system-level description to a component-level specification. The proposed framework assists the IPT in

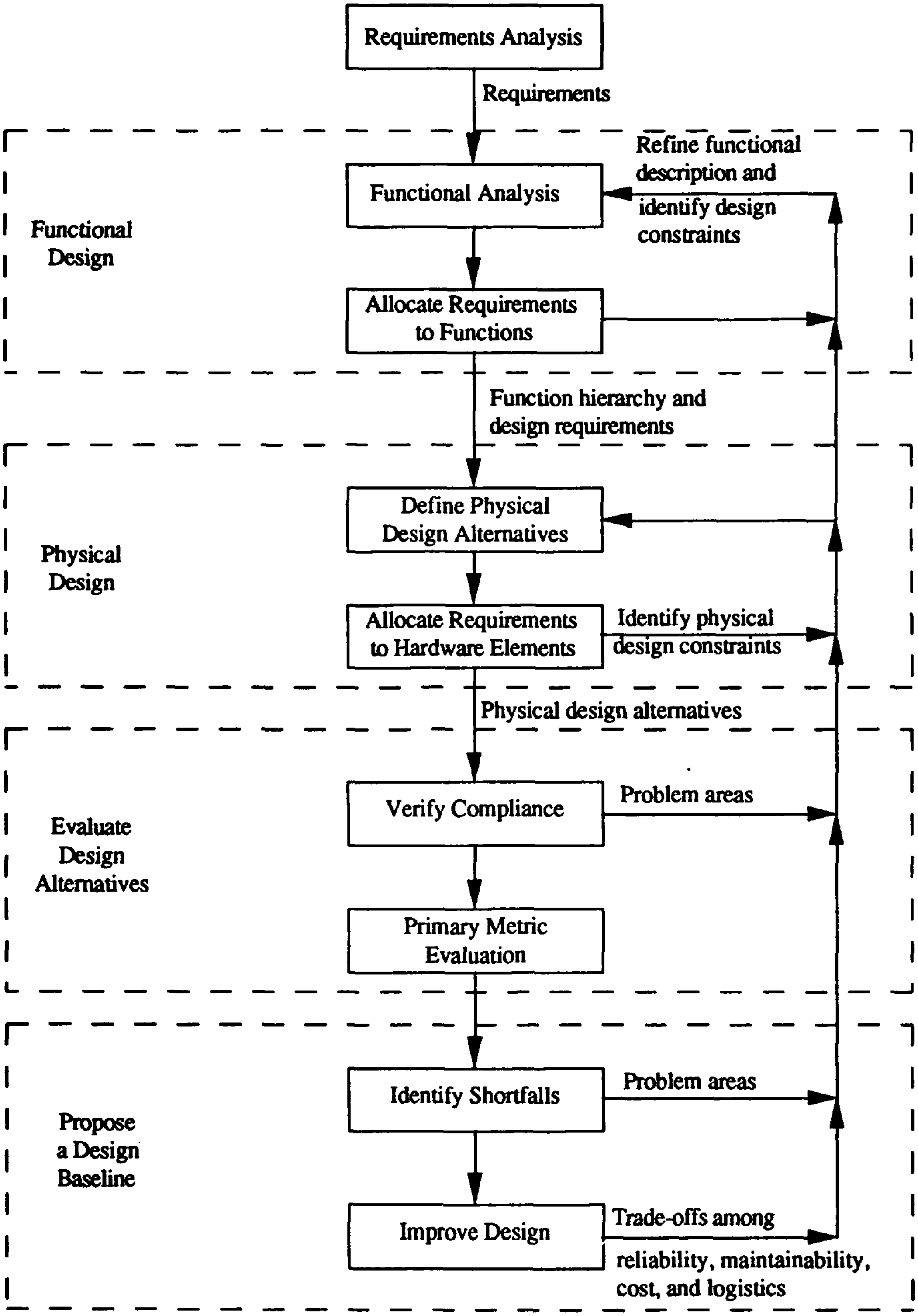

Requirements Analysis
Requirements
Functional Design
Functional Analysis
Refine functional description and identify design constraints
Allocate Requirements to Functions
Function hierarchy and design requirements
Physical Design
Define Physical Design Alternatives
Allocate Requirements to Hardware Elements
Identify physical design constraints
Physical design alternatives
Evaluate Design Alternatives
Verify Compliance
Problem areas
Primary Metric Evaluation
Propose a Design Baseline
Identify Shortfalls
Problem areas
Improve Design
Trade-offs among reliability, maintainability, cost, and logistics

Figure 1
Basic Proposed Design Process

integrating earlier design activities that focus on defining, apportioning, and allocating resources and requirements for the equipment to be designed with later activities that have as their focus the design or identification of specific hardware and software implementations that satisfy the requirements.

It is generally not possible to depend wholly upon formal optimization methods to develop an optimal design. Consequently, efforts at optimization should revolve around trade study evaluations of as many design alternatives as the IPT can identify and investigate with the resources and time available. For the time being, the identification and evaluation of design alternatives should remain a task left to the experience of the IPT members because both of these tasks depend heavily on the collective knowledge and creativity of the IPT.

A key element in this method is the derivation and prioritization of the functional, performance, and RM&S requirements and objectives. The IPT needs this information to provide a basis for determining the relative importance of specific design parameters and attributes, and for identifying design trade-offs. Trade studies and similarity analyses[2] are the primary means of determining the characteristics of the design that best meet the design requirements. For the process to be responsive to the design requirements, the IPT must characterize functions in terms of attributes such as purpose, complexity, and criticality. Another key element is the use of sensitivity analyses to identify the design characteristics and areas of the design that have the greatest impact on the design attributes. The IPT needs the results of these analyses to identify the areas of the design which, if modified, would yield the most cost-effective improvements and to assess the implications of proposed design modifications.

The inputs to the design process include a description of the functions the object is to perform, a definition of the inputs and outputs of the object, a description of the physical interfaces between the object being designed and other objects with which it interacts, and a definition of the design requirements applicable to the object and its interfaces (e.g., external signal, reliability, maintainability, physical dimension, and cost requirements).

The proposed process assumes, for example, that the IPT has performed a mission analysis or use study to establish the basic operational, maintenance, and support requirements. The process also requires the availability of a functional description of the unit at the system level (i.e., a description of the required functions and their relationships), a definition of the external interfaces

[2] The process of comparing similar proposed and existing designs to develop estimates for design attributes of interest (e.g., RM&S and LCC) is referred to as a similarity analysis.

of the unit (i.e., an Interface Control Document or equivalent), and a relatively comprehensive definition of the design requirements and criteria to which the unit will be designed.

Although shown as separate, sequential tasks in Figure 1, the design process tasks are highly coupled and, in practice, there should be considerable overlap and feedback between adjacent tasks. Each of these tasks is discussed below in more detail.

Requirements Analysis

The first step in the design process is to derive a comprehensive definition of the requirements the design must satisfy. In addition to requirements specified by the customer or allocated from previous levels, the IPT needs to enumerate requirements derived from analyses at the current level (e.g., a reliability or testability analysis) or introduced by design or manufacturing standards. To facilitate subsequent analyses, the IPT should document the type, source, quality of the source, and basis of each requirement (e.g., "customer requirement" or "derived from unit-level analysis"). Making this information available throughout the remainder of the design process is critical to the success of the design process. This information should form the basis for many choices the IPT makes during subsequent trade studies and alternative analyses.

To focus the design process, the IPT needs to establish the relative importance of the design requirements and define the design parameters and evaluation metrics (i.e., measures of effectiveness) that will be the primary basis for evaluating the design. The IPT should rank requirements and metrics along several dimensions: (1) the priorities of the customer; (2) the relative impact a requirement or metric is expected to have on the performance, quality, etc. of the design; (3) whether each requirement is a constraint or a goal; and (4) the perceived difficulty in meeting each requirement. The technique used to develop the ranking is not particularly important to the outcome of the process provided the ranking represents the team consensus. The IPT should then merge these lists into a single prioritized list, which may group requirements rather than assigning a different importance to each. Again, the particular technique used to merge the lists is not important as long as each member agrees that his/her concerns are represented.

The key to a successful outcome is to get each IPT member involved in the process and to ensure those with contrary views attempt to identify a solution all can support. If the IPT cannot agree on a single, unified list (i.e., becomes "deadlocked") then several lists should be developed that represent the extremes posed by the alternative viewpoints and these should be examined by members of an expanded team. If all fails and the IPT remains deadlocked, a facilitator (i.e., someone skilled in group decision-making) should be brought in and the impact of the alternative prioritizations should be investigated. It is often the case that someone skilled in facilitating group decision-making can help the IPT develop a consensus or, barring an agreement on the requirements ranking, help the group identify a design strategy or design that satisfies alternative rankings equally well.

Although there is no single correct way to develop this prioritization which is applicable in all, or even most, instances, it is possible to develop heuristic guidelines that apply in many instances. For example, the IPT should consider requirements establishing ambitious acceptance criteria—values or tolerances exceeding historic levels or instances in which few values are acceptable—as primary evaluation criteria and place them high on the list since they are likely to drive the design.

Specialty engineers should play an active role in identifying and prioritizing requirements. This role includes identifying requirements that design engineers may have overlooked and recognizing instances in which requirements in their area of specialization are likely to be difficult to meet, either by their very nature or due to their impact on requirements in other areas of the design.

The design requirements should then be recast as design constraints and goals. The type of goal or constraint—global or local—should be established since the approach to satisfying them differs. Global design goals and constraints are measures of the total weight, performance, etc. of the unit, while local goals and constraints apply to only some elements of the unit (e.g., a response time requirement associated with a particular function). Local design goals and constraints can be directly allocated to the appropriate unit element while global goals and constraints must be apportioned before being allocated. If dependencies exist between individual requirements, such as a customer-defined relation between unit cost and performance, they should also be recast as design constraints and goals.

Where possible, the IPT should establish intermediate design goals (both local and global) to guide the search for a design solution. In all instances, the type, source, basis, and other pertinent information should be noted and made available to the design engineer for each design requirement and metric. As previously noted, this information is the basis for the choices the IPT makes during the subsequent trade studies and alternative analyses.

Functional Design

The purpose of the functional design task is to develop a detailed functional description of the item (e.g., definition of functions, functional block and flow diagrams, definition of interconnects, and apportionment and allocation of requirements to subfunctions) sufficient to support physical design at the current level of detail, and further functional design at a level of greater detail. The approach is to refine the functional description from the previous level by decomposing each function into a set of subordinate functions. During this task, the IPT identifies critical functions and signals, and apportions and allocates the requirements from the previous level among the functions at the current level. If solutions have not been dictated by the results of analysis at higher levels of abstraction, the IPT also proposes and evaluates alternative functional architectures and partitioning, packaging, fault tolerance, and testability concepts.

The above process should continue until the IPT identifies one or more physical components that are capable of providing each function. The product of this task should be a functional hierarchy in which the intermediate levels of the hierarchy represent intermediate decompositions of the functions and the leaves of the hierarchy (i.e., the lowest level in the design hierarchy) are the functions for which the IPT has defined one or more physical implementations.

Functional Analysis

Functional analysis includes two major types of design decisions. First, the IPT determines the criticality of each function required of the item. This criticality is important to apportioning and allocating resources to design tasks and choosing among competing alternatives during the design process. Functional analysis also includes functional decomposition, in which the IPT decomposes each function required of the item into more detailed functions until a function can be realized directly in hardware.

Function Criticality. Before the IPT can determine the function criticality, it must establish a mission profile or duty cycle for the unit being designed and perform a mission analysis to establish the minimum operational configuration of the unit (i.e., the minimum set of functions required to complete or satisfactorily perform the primary mission of the unit). If optional missions are being proposed for the unit, the IPT must identify the functions needed to support only these missions and not the primary mission. Once such distinctions have been determined, the IPT can define the role of each function of the item and establish its criticality according to the levels described in Table 1. This list, an expansion of that normally used in industry today, is needed to better discriminate the impact of losing a function on the operation and maintenance of the item.

For example, a design engineer would classify a function whose sole purpose is to provide an on-board test capability for maintenance fault-isolation as having a "Maintainability Degradation" criticality, and define its role as "on-board test for maintenance isolation."

Table 1. Function Criticality

Criticality	Description
Safety Critical	Loss of function will cause death or severe injury
Equipment Loss	Loss of function will result in loss of or damage to equipment (mission completion is prevented)
Mission Critical	Loss of function prevents mission completion (no loss or damage to equipment)
Performance Degradation	Loss of function degrades mission performance and effectiveness
Maintainability Degradation	Loss of function degrades maintenance effectiveness
Noncritical Function Degradation	Loss of function degrades fault-tolerance (repair at earliest maintenance opportunity required)
Negligible	Negligible impact on the mission or support

Functional Decomposition. There is significant interplay between the functional and physical aspects of the item that the IPT must consider during functional decomposition. This interplay affects the decomposition in two ways. First, the cost and feasibility of implementing specific functions may shape the decomposition by ruling out some alternatives. Second, each feasible alternative can have special attributes which further affect the decomposition. For instance, the IPT's choice among digital, analog, or mechanical implementations will significantly affect the functional decomposition of the modem function mentioned in the earlier example. Modem functions that can be implemented using one technology may be unnecessary, too costly, or infeasible in another. Similarly, the physical design of a function may introduce the need for additional subordinate or supporting functions. A function implemented as a set of discrete components may require an isolation buffer or other circuitry to integrate components that would not be required if the function were implemented by a single custom component. As the functional decomposition progresses, the IPT should establish the role of each newly defined function, determine its importance to the mission, and assign it a criticality according to the levels in Table 1.

Design engineers should be in charge of the functional decomposition of the design, and physical design specialty engineers (e.g., packaging engineers) should be in charge of its physical definition. Because of the interplay between physical and functional design, these groups, as well as other specialty engineers, should participate in regular and frequent meetings to ensure, to the greatest extent possible, the choices posed by Table 2 are addressed.[3] These design review

[3] Table 2 lists the key design choices at each level in the design hierarchy that, to a large degree, determine the RM&S of a design. For example, at the system and subsystem level, the architecture and the interconnect design chosen by the team dramatically affect the reliability of the system while the level of system health monitoring chosen by the team significantly affects its maintainability.

meetings should be structured so that specialty engineers have an opportunity to identify and assess the impacts of proposed functional decompositions on their area of expertise as they evolve, thereby avoiding any unforeseen impacts of a proposed decomposition. In particular, the IPT, and especially the specialty engineers, must assess the impact that choices in each area mentioned in Table 2 will have on the item and its RM&S.

The meetings should also provide all IPT members with opportunities to become knowledgeable of the other specialty areas and to identify where choices and implementations have proved to be felicitous or undesirable. The IPT should also make a concentrated effort during these meetings to recognize and catalog instances where attractive alternatives were ruled out because of problems stemming from one or more specialty areas. This will serve to identify attractive areas for research aimed at giving the design engineer greater flexibility in the future. Finally, the IPT should include a specialist with experience in such sessions. This specialist should be tasked to condense the "lessons learned" and collective knowledge of the team into heuristics and guidelines for distribution to its members and other IPTs.

Allocate Requirements to Functions

Once the IPT has completed the decomposition of a function to the next level of detail, it should apportion and allocate the requirements for that function among the subfunctions of the decomposition. To support this process, the IPT must have or develop a mission or performance model that reflects the architecture and reliability of the functions at the lower level. The IPT can use this model not only to determine the reliability and performance allocations, but also to evaluate the sensitivity of the item to a variety of architectures and to the reliability and performance of individual design elements. If the decomposition requires reliability or performance levels that exceed historic levels, the IPT should investigate alternative architectures and hardware implementations of specific functions.

At this point in the design cycle, the physical design is, in general, not well defined and, as a consequence, detailed maintainability and supportability requirements cannot normally be apportioned and allocated to individual functions. Usually some level of physical design must be completed before the maintainability and supportability requirements can be apportioned and allocated. The few exceptions are those cases in which the physical implementation of the function has been determined.

Table 2. Reliability, Maintainability, and Supportability Design Matrix

Design Level	Reliability	Maintainability	Supportability
System, Subsystem	• System architecture and fault-tolerant features (e.g., reconfiguration and unit redundancy) • System operating environment • System interconnect design (e.g., number, design, and quality of interconnects and cables) • Mechanical design of chassis and chassis-unit interface	• Maintenance concept (e.g., 2- versus 3-tier maintenance and deferred maintenance) • System reliability • System health monitoring • System integrated test features and effectiveness • Maintenance procedures and equipment (e.g., need for automated test equipment [ATE] and test sets) • Level of repair and discard • Unit access for inspection, preventive maintenance, and interchange • Unit installation complexity and tools needed • Maintenance crew size	• System concept • System reliability and maintainability • System modularity and commonality • Level of repair and discard • Availability, number, and cost of spares • Facilities and support equipment • Manpower requirements
Unit (LRU, LRM)	• Unit architecture and fault-tolerant features (e.g., partitioning and function redundancy) • Operating environment (e.g., temperature, and electrical and mechanical stresses) • Number, design, and quality of component connections • Component and connector mounting to board	• Health monitoring features and effectiveness • Integrated test features and effectiveness • Test procedures and equipment (e.g., need for ATE and test sets) • Repair versus discard • Internal access for inspection, preventive maintenance, test, and repair • Assembly/disassembly complexity and tool needs • Human factors (e.g., physical dimensions, weight, and center of gravity)	• Number and cost of spares • Spares availability (e.g., number of manufacturers, component availability of spares) • Repair versus discard • Unit modularity and commonality (e.g., form factor and interoperability) • Component and connector interchange needs (e.g., equipment, skill, and tools)
Function, Circuit Board	• Fault-tolerant features (e.g., redundancy, masking, and containment regions) • Failure modes • Electrical and thermal stress levels • Circuit sensitivity to noise and component variability • Integration level	• Circuit testability (e.g., number and complexity of component failure modes, controllability, and observability) • Built-in test (BIT) built-in test equipment (BITE), and built-in self-test (BIST) features and effectiveness • Calibration and trimming requirements • Integration level	• Circuit commonality • Circuit complexity (e.g., training requirements)
Component (component selection)	• Component failure rate and modes • Component quality (e.g., manufacturing variability and thermal sensitivity)	• Component failure rate and modes • Component testability and BIST • Component durability	• Component failure rate • Component availability • Component commonality • Component durability

Physical Design

The preceding step should yield one or more functional design alternatives warranting further investigation. During the physical design step, the IPT explores alternative physical implementations of each function or functional group. The initial focus of this effort should be on those functions needed to perform the primary mission. This will ensure that the design satisfies the requirements of the primary mission before the IPT adds additional functionality (and, therefore, additional hardware elements) to meet the requirements of any optional missions. Evolving the design in this manner will optimize it for the primary mission.

Define Physical Design Alternatives

The nature of the physical design alternatives varies with the level of the design. At the higher functional design levels, the alternatives will be such things as structural material, manufacturing process, partitioning and packaging of design elements, etc. If possible, the IPT should consider several alternatives it believes will satisfy the design requirements. The alternatives should cover or bracket options considered appropriate for the design problem. Usually, the design problem, development schedule, available engineering resources and budget, and level of risk the IPT is willing to accept limit the number and scope of design alternatives that can be investigated. Other constraints, such as a requirement to meet specific radiation or environmental factors or those associated with the use of a specific technology (see Appendix B for examples), also limit the alternatives that can be considered.

At all but the most detailed levels of design, evaluation of candidate design alternatives is based on similarity analyses. These analyses will be most useful if the IPT derives at least one alternative in each major class (i.e., design approaches such as custom integrated circuits [ICs] and off-the-shelf components) from existing designs or design elements and if each alternative is known to provide the needed functionality. The IPT should use the analyses to determine the aspects of the design—design characteristics, parameters, or areas of the design—that have the most influence on the development schedule, product quality, manufacturing process, and product supportability. A key goal of the analyses should be to identify the design aspects that will drive the design process and determine the sensitivity of the design and design process to these attributes.

Experts in physical design, reliability, testability, and other areas should participate in these studies by performing those parts of the similarity analyses that are in their area of expertise and critiquing the analyses results. The IPT should meet regularly to consider the results of each analysis and choose among the proposed alternatives. As is the case for functional design, the IPT should include specialty engineers from all design areas, whether or not studies from their area of expertise are among those being considered, and seek to effectively utilize and improve the collective knowledge of the team. By involving engineers with a variety of perspectives in identifying alternatives and choosing among them, the process increases the possibility that a broader range of alternatives will be considered.

Allocate Resources and Requirements to Hardware Elements

Once it has defined a physical implementation for the functional design, the IPT should apportion and allocate the resources and requirements among elements of this physical implementation. The IPT should employ an approach similar to that used during functional design. Again, the allocations should reflect performance and RM&S levels that are believed to be achievable. To insure this, the complete IPT, with representatives from all of the specialty areas, must be involved in the apportioning and allocation process.

Evaluate Design Alternatives

In this portion of the effort, the IPT evaluates the design alternatives derived in the preceding step. The goal of the evaluation process is to not only determine how well a proposed design alternative satisfies the design requirements, but also to reveal the relative strengths and weaknesses of each alternative, and to identify which of the proposed alternatives best satisfies the design requirements and objectives.

The evaluation process consists of two major tasks: (1) checking the design for compliance with the applicable design standards, practices, and guidelines; and (2) comparing the values of the primary evaluation metrics with the allocated or required values. The evaluation should involve both quantitative and qualitative comparisons of the design characteristics and measures of effectiveness and, where possible, account for the basis and quality of individual estimates. Finally, the evaluation results should include a confidence band for the key design parameters and evaluation metrics based on a consideration of their basis, accuracy, and uncertainty. A sensitivity analysis can provide the basic data needed to determine this confidence band.

Verify Compliance

Each IPT member must verify that the design complies with the design standards, practices, and guidelines in his/her specialty area, such as those in Appendix B, as the design evolves. A review of Appendix B shows that most of the guidelines are applicable only at a particular time in the design process (e.g., systems design or unit design), apply only to particular design elements (e.g., circuit board or component), and frequently address the concerns of only one or two of the specialty areas. Consequently, verification of compliance requires review of the design by representatives from the various specialty areas.

Primary Metric Evaluation

The information developed during the requirements and functional analysis steps is a vital element of the evaluation process. The type, source, basis, and relative importance of the requirements; the importance, role, etc. of each function; and information about the quality and source of the estimates are critical to determining how well an alternative satisfies the requirements and whether one alternative is superior in all respects to the others. This information makes it possible to order the design elements according to the importance of the functions they perform and the requirements they address. Although this information will not guarantee that an optimum solution to the design problem is found, it will allow the highest rating to be given to the design alternatives contributing to the most important functions and satisfying the most important requirements.

One danger in using a simple evaluation function to evaluate a design is that significant variations in one primary evaluation criterion may be "washed out" and go unnoticed. As noted in Section II, individual design requirements should be established and met for many of the design criteria involved in the design of electronic systems.

Propose a Design Baseline

Once the IPT completes its evaluation, it must select a design baseline from among the proposed design alternatives or synthesize one from elements of one or more of the alternatives. This baseline will serve as the foundation for any subsequent design activities.

Identify Shortfalls

Once a baseline has been established, the IPT must identify the design requirements and criteria that are not met or are only partially met (i.e., the requirements shortfalls), the source of each shortfall (i.e., the function, module, or component), the reason for the shortfall, and the importance of the shortfall to the quality of the design. In addition, the IPT should identify the design parameters or attributes and the components or areas of the design that have a major impact on key product characteristics—the major "design drivers."

These design drivers can be determined by performing a sensitivity analysis to identify the parameters and areas of the design that have a large influence on the key design attributes. For example, the failure rate of selected critical functions or design elements can be varied to determine the impact of a change in reliability on the mission completion success probability (MCSP) or other mission reliability metrics.

Improve Design

If none of the existing design alternatives adequately satisfy the requirements, the IPT has four options:

a. select one design alternative and seek a relaxation in the design requirements or criteria that are not satisfied,

b. modify one design alternative to rectify the shortfall(s),

c. devise another alternative from elements of the proposed design alternatives, or

d. propose an entirely new alternative.

The first option is available only if most of the requirements are satisfied and if the unsatisfied requirements are considered "soft." If the requirements at issue cannot be relaxed, the IPT has no choice but to select one of the other three alternatives. The IPT should explore the alternatives in the order listed, which is the order of increasing cost and difficulty of implementation. Additional specialty engineers should participate in this process. In many cases, the shortfall(s) will be in a metric associated with one specialty area. For example, the shortfall(s) could be in MCSP, LCC, or a maintainability or testability metric. These cases usually require the expertise of the specialty engineer from that area to correct the shortfall(s).

The IPT should identify candidate changes to a proposed design and rank them according to the number of shortfalls they address, the type and importance of each shortfall, the potential improvement in the shortfall the candidate modification will yield, and the projected impact of the candidate modification on other areas of the design.

Once all requirements are satisfied, the IPT should explore fruitful areas for design improvement, as identified during the sensitivity analysis described in the previous section, as potential areas for design optimization. In most cases, this optimization will still depend on a "generate and test" approach for which the IPT proposes and tests each new possibility separately or in combination with others. The process is not formalized and there is no assurance that it will identify the optimum solution to the design problem. The generation of possibilities is conducted by a group of design engineers but a broader group, including the relevant specialty engineers, should frequently review the progress and results of this task.

As part of the RAMCAD research, a system entitled SIDECAR was developed to automate the exploration of design alternatives for an electronic design problem. This program can assist the IPT in estimating the reliability of electronic designs and determining the contribution of each element in the design to the overall system reliability. SIDECAR can also assist the IPT in exploring the sensitivity of the reliability, cost, etc. of an item by replacing a function or component with elements of equivalent functionality that are drawn from its database of designs and components. This prototype system is described briefly in Section IV and in more detail in other reports (Yount & Siewiorek, 1991; Tracy et al., 1993).

In isolated cases, it may be possible to parameterize a design problem (i.e., explicitly represent the relationships of all design parameters and criteria of interest by a formula). In such cases, it may be possible for the IPT to devise a suitable evaluation function and to use formal design optimization approaches to find the "optimum" design. Note that this design is optimum in a limited sense: it is the optimum design with respect to the given evaluation function among the designs represented by the parameterization.

IV. SUPPORT REQUIREMENTS

The success of the methodology proposed in this report depends on several types of support. Much of this support is needed for the design methodologies currently used in industry; however, in some cases the emphases in the proposed methodology differ from other approaches. In this section, the general support needs are presented and the opportunities to provide computer support for the methodology by addressing these needs in an automated design environment are discussed.

General Requirements of the Methodology

The proposed methodology is based on a process of evaluating a representative set of all possible approaches to a design problem. The success of this approach depends on how well the alternatives considered by the IPT cover the range of possibilities. Also, choosing among the alternatives requires evaluations of various design attributes (e.g., RM&S and LCC). Since the alternatives are not defined in sufficient detail to support a direct analysis early in the design process, this information is necessarily derived from similarity analyses. To ensure the comparisons produce representative information for the evaluations, specialty engineers from a variety of areas must participate in these analyses to ensure that the metrics in their specialty area (e.g., RM&S and LCC) are correctly estimated. The design engineers, in turn, are responsible for estimating attributes such as unit cost, performance, and weight. Finally, it is important that design organizations keep all engineers informed of the field performance of their designs so this information can influence their approach to design problems.

Coverage of Possibilities

Several approaches can be employed to make it more likely that the possibilities considered by the IPT adequately cover the potential design space. The most obvious approach is to involve a group of engineers with a wide range of experience in the sessions devoted to identifying candidate designs. Where possible, this group should use brainstorming techniques and tools (e.g., outlining tools and group decision support tools) to enhance its effectiveness. Because not all members of the IPT can participate in all sessions (due to scheduling conflicts, resource limitations, etc.), it is imperative that the session discussions and decisions be amply documented for those that could not attend. Not doing so often nullifies the potential advantages of such sessions. Brainstorming sessions monitored as part of this research indicate that brainstorming sessions which include computing aids are three to four times more productive than those which rely on human scribes.

A second approach is to employ tools such as the SIDECAR program to aid the IPT in automatically exploring the design spacε associated with a design problem. Such tools allow the IPT to define the range of possibilities that should be investigated and to alleviate the need for engineers to manually compose each of the alternatives.

It is also important that group members become familiar with new design approaches. Design organizations should devote some of their resources (e.g., time, personnel, and computing resources) to researching both attractive new techniques culled from the literature and locally developed ideas. Without such a research commitment, it is unlikely that design engineers will be comfortable employing new approaches when faced with an actual design problem.

Similarity Analysis

Even if a set of design alternatives cover the design space perfectly, the proposed methodology will fail if the IPT cannot acquire the design information needed to evaluate each alternative. As noted in the introduction to this section, until the design is sufficiently defined to enable a direct evaluation, evaluations are necessarily based upon similarity analyses, preferably using fielded designs. The success of these analyses will depend on the aptness of the existing designs chosen as the basis for the analyses and on the ability of the IPT to adjust the evaluations for differences between the existing design and the current design alternatives. Thus, the concerns addressed above in the discussion of coverage of possibilities also apply to the similarity analyses. The broader the collective knowledge and experience of the IPT, the more likely it is that an appropriate design will be chosen as the basis for the analyses.

Given a perfect choice of a design on which to base the analyses, the IPT must accurately determine which differences between the base design and the current design alternatives are significant, and how to account for them. To increase the IPT's effectiveness in this area, there must be a consistent process to compare estimates with more accurate evaluations when they become available. These comparisons must be used to improve the estimating abilities of all IPT members, not to evaluate those preparing the estimates.

In addition to brainstorming and other group decision support tools, spreadsheets and tools such as SIDECAR are needed to assist the IPT in performing the similarity analyses. Computer supported sensitivity analyses provide a mechanism to determine the significance of differences between various design attributes to the overall comparison.

Involvement of Experts

The proposed approach is based on the use of an IPT (a team composed of design and specialty engineers from each major discipline). An underlying assumption of this approach is that such a team can more readily identify and correct problems early in the design process than a team not having representatives from each discipline, thereby resulting in significant savings by avoiding problems that would require redesign. However, design engineers currently perform the brunt of the early design work since many of the specialty engineers require information that is only available once a design has been proposed. Currently, the primary role of the specialty engineers early in the design is to propose the basic concept for their area (e.g., manufacturing concept), to identify any design constraints in their area (e.g., use of a particular manufacturing process), and to critique and evaluate proposed design alternatives. Consequently, current multidisciplinary design teams may add significantly to the cost of the design without a commensurate improvement in the quality of the design unless the teams are well managed, design information is widely available when needed, and the results of the specialty engineers' analyses are disseminated to the team in a way that enables the nonspecialist among them to understand it.

The proposed approach introduces additional computing requirements above and beyond those needed to support a single design engineer. Computing aids are needed to help the IPT function effectively. For example, computing aids are required to assist the IPT in coordinating the members' activities (e.g., scheduling tools and electronic mail), disseminating and presenting design information (e.g., design databases), and conducting effective meetings and brainstorming sessions (e.g., design rational capture tools and group decision support tools).

Feedback from the Field

One major weakness in the design process that the RAMCAD research revealed is the lack of feedback from the field. Often there is no formal process to inform design engineers of the performance of their products in the field, and little informal flow of such information. There are several reasons for this. Two of the more significant reasons are the difficulty of collecting accurate field data and the difficulty of locating the responsible design engineer when such data become available.

At present, accurate field data is difficult to collect and disseminate for several reasons. The main reason is that the collection and reporting of such data is time consuming, error-prone, and peripheral to the task of maintenance personnel. In many cases, maintenance organizations do not have the time and resources required to consistently produce high quality data. The lack of a consistent high quality source of this data is a major barrier to the effective use of field experience in the design of new systems. To a large degree, this problem can be addressed by automating the field data acquisition process. In addition to incorporating improved BIT and BITE into existing systems to better isolate the source of malfunctions, computer scannable identification labels could be attached to components. Maintenance personnel could then identify and collect the serial number, component type, and other important maintenance data pertaining to a faulty component or replacement component by electronically scanning this label. This would greatly simplify the task of identifying which parts were replaced during a maintenance action. Electronic scanning could also greatly simplify the part ordering process, providing maintenance personnel with both the ability and the incentive to collect accurate data.

A second obstacle to collecting accurate field data is the difficulty of accurately determining the cause of a fault. At the first maintenance level, the goal is usually rapid turnaround. If field testing cannot unambiguously identify the malfunctioning unit, maintenance personnel must replace suspected units, one at a time or all at once, so that system operation is restored. In this process, it is quite common to replace several units without any fault and sometimes the source of the fault is not even identified. To get useful field data back to the IPT, the true source of each fault must be identified and functional units that have been replaced but that are not actually faulty must be identified and removed from the database of faulty units. Although scanning and other means of automatically identifying parts and units can simplify this process, the maintenance organization must go beyond identifying suspected faulty parts and commit the resources necessary to identify the fundamental cause of the fault and ensure the status of all replaced units is reported correctly.

One example of how this should be accomplished can be found in a problem solved by the Boeing Commercial Airplane Group. A thorough analysis of the faults found in a set of electronic parts showed a significant percentage of the detected faults were caused by small amounts of oil in a sensitive area of a single part provided by one supplier. With this knowledge, that supplier's manufacturing and quality control processes were identified as inadequate and needing improvement. The majority of the remaining faults were determined to be caused by other manufacturing processes—chiefly bent pins. Clearly these data are valuable to the design of new systems, since the data show that the design of the system is not the primary source of the problems observed in the field (i.e., the inherent reliability of the basic design did not need to be improved). Instead, the basic design could be used in future systems but the problem supplier should be avoided until the problem processes are improved. However, additional hardware resources might be added to the basic design to verify pin connections, or an improved connector design or manufacturing process may be warranted.

Even if accurate field data are available, it is difficult, particularly on military projects, to direct the data back to the actual design engineer who created a fielded part because of the nature of military design work. Usually, the design work is completed and the design team disbanded before significant quantities of the system are produced, let alone fielded. Except for a team of sustaining engineers, the design engineers responsible for the design of a system are no longer involved with the project and have moved on to other projects, possibly in other companies. To address this problem, the specialty engineering organizations (e.g., reliability and maintainability) must persist between projects to maintain, analyze, and disseminate field data to any ongoing design efforts. To facilitate the communication between these organizations and the active IPTs, these organizations must develop and maintain a computer database containing the results of their analyses of fielded systems, particularly those determined to be significantly different from the norm (i.e., highly reliable or unreliable, or very easy or difficult to maintain).

An integral part of the proposed methodology is a design database in which the design, the decisions made in the course of producing a design, and the evaluations on which those decisions were based are captured. With the aid of such a database, the specialty engineering organization could relate field experience back to the relevant design characteristics and decisions. The continual involvement of specialty engineers in the evaluation of design alternatives and fielded problems will provide a means for the results of field experience to impact future designs. The database should consolidate such information and provide access to this information based on the attributes and features of a proposed design. Doing so will support the similarity analysis task described above as well as design modification tasks. In addition, such a database of field data can assist the specialty engineers in identifying areas in which research is needed into new design and manufacturing techniques or improved estimation and prediction methods.

Computer Support Requirements of the Methodology

The computer support requirements discussed in the following subsections are needed to enable an effective application of the proposed methodology. However, their implementation would improve the design environment and process even if the proposed methodology is not employed. These requirements are in addition to other commonly discussed requirements (e.g., see Birmingham et al., 1988) and cover four general areas. First, a design repository is needed that can support the entire design process, thereby enabling information developed at one stage of the design to be easily accessed and employed at subsequent stages and times. Second, methods and tools for estimating design parameters need to be improved to make the estimates more accurate, informative, and available. Third, the design environment should provide the design engineer with on-line access to design guidelines and support for checking the compliance of a design with these guidelines. Fourth, the environment should provide a database for recording design alternatives and choices, and the basis for these choices.

Design Repository and Representation

IPTs need a common design repository that supports the design process described in Section III and integrates the design perspectives of each discipline in order to facilitate interaction among the IPT members. Clearly, no existing, single representation of the design will adequately support all the needs of the engineers at all levels of the design and during all design phases (e.g., the needs of a system engineer and a circuit design engineer vary significantly in the data and level of detail each requires). To provide effective support, the design environment must provide the means to capture and present the design information to each engineer in a way that can be customized to the individual's needs. To support the collaboration of the IPT and an evolving design, the design representation underlying each of these perspectives must integrate the decision support models used in the early phases of the design with the representations required for system and detailed design.

In addition to documenting the design, the design repository must actively support the design process. The design environment must support the decomposition of functions and the apportionment and allocation of resources and requirements to successively more detailed levels of the design as well as the definition and evaluation of alternative physical implementations. The environment should also support the processes of verifying that a design complies with applicable design practices and standards, and that allocated requirements are reasonable and do not exceed maximum permissible values. As the design progresses, and parameters are evaluated at more detailed levels, the system should also assist the IPT in comparing new estimates to previous and expected values, and in flagging instances for which these estimates combine to exceed allocated requirements. Clearly, if the requirements at a given level are consistent with system requirements, then the combining of estimates at lower levels in the design hierarchy should not exceed the system requirements unless at least one of the lower level requirements exceeds its allocation. Because newer, refined allocations and estimates routinely differ from previous values in both directions (i.e., each allocation is either increased or decreased), a system is needed to monitor these changes and inform the IPT of a change that is significant (e.g., exceeds a predefined threshold or vastly alters other requirements due to an interaction between the requirements). The system that monitors these changes should also track the combination of all of the newest, and presumably most accurate, estimates so the IPT can determine when a combination of variations is significant in one direction or the overall value exceeds the requirements and must be addressed.

To a large degree, the technology to develop such an environment exists today. This section includes a brief description of a prototype system developed as part of this research effort, the SIDECAR system, that could provide some support for the process described above.

Improved Estimates of Design Parameters

Currently, design engineers cannot obtain timely estimates for many of the design parameters and metrics described in Section V during the early and middle phases of design synthesis. However, it is during these phases that design engineers need this information to help them properly design a system. There are two principal reasons for this lack of information: (1) preparing the estimates is a lengthy process that often requires the services of an expert and (2) the estimation techniques often require detailed information that is not available early in the design process. A design support environment could provide the design engineer access to several alternative methods for estimating the value of these metrics, and to guidelines that describe how meaningful estimates can be derived from similar designs and high-level attributes of the current design. Specialty engineers should develop, codify, and maintain this information, and develop "rules of thumb" and other methods the design engineer can employ to verify that the computed values are reasonable. When the estimates prove in error or inaccurate, the system should provide an easy means for the design engineer to notify the appropriate specialty engineer and provide that specialty engineer with the information needed to determine the cause of the inaccuracy. When the specialty engineer corrects or revises the guidelines and methods to provide better estimates in the future, the system must provide a means of notifying the affected IPTs.

As the design becomes more detailed, the environment should provide design attribute estimating techniques that are based on a direct analysis of the current design and its parts instead of similarity analyses. In cases where an algorithm is known for computing the estimate, the process should be automated to allow the design engineer access to computed results on request. In such situations, the environment should compute and display the sensitivity of the computed estimate to its inputs and assist the design engineer in making the appropriate changes to bring the value of an estimate within a required range.

In addition to estimates, the design engineer needs information about the quality of these estimates and other design data within the design information system. The environment should provide the design engineer with information that identifies the quality of the estimate, the quality of the data used to compute it, and the nature of the process used to compute it. This information will help the design engineer choose among alternative approaches, provide contingent information about the quality of the resulting estimate, and help the design engineer determine when a change in the value of one or more input parameters justifies recomputing an estimate. Finally, the environment should automatically "roll up" estimates, compare the results to the requirements, and warn the design engineer when there is significant overshoot or undershoot. This is another area in which the information about the quality of the estimates is important because it indicates when a divergence from the requirements is significant.

The SIDECAR system described later in this section provides routines to assist the design engineer in determining the sensitivity of an estimate to variations in the data from which it is computed and in assessing whether a given metric value is achievable for a modeled design (a marginal benefits analysis). It does not perform the information quality analysis described above.

Access to Design Guidelines

Most design organizations have defined standards and guidelines to ensure that designs produced by the organization meet minimal quality levels. Usually these standards and guidelines are found in multiple-volume paper documents which are, at best, difficult to use or, at worst, totally unusable by the design engineers. Because they are paper documents, the timely distribution and maintenance of this information is also a daunting task. In the proposed design environment, these standards and guidelines would be available on-line, with flexible indexing to enable a design engineer to identify, locate, and retrieve any standards or guidelines relevant to a particular issue or design problem.

This concept presents several research challenges that must be overcome before it can be implemented. One is to develop the capability to automatically display standards and guidelines relevant to a design based on the attributes and features of the design. The ability to do this rests on the use of a standard representation of the guidelines and standards, and the development of a design classification taxonomy. If sufficiently detailed semantics can be associated with particular design attributes and features and with the design as represented in the environment, it may be possible to associate standards and guidelines with particular aspects of the design and automatically display them when they are relevant.

Another challenge is to compile design rules relevant to a particular specialty area and integrate them into the design environment in such a way that the design engineer receives guidance when modifying a design to satisfy the needs of that specialty area. Appendix B contain examples of the type of information that must be incorporated into the design environment to support some of the specialty areas. The simplest approach is to encode this information as an on-line hypertext document that is customized by the specialty engineers to specific classes of design problems. A more ambitious goal would be to incorporate the information with sufficient semantics in the representation to enable, for example, an automated system to apply the rules to a design being modified to meet testability requirements and inform the design engineer when a desired testability structure is being omitted or when design elements that are undesirable from a testability perspective have been included.

An attempt should be made to develop similar collections of rules and guidelines for each specialty area, although this may not be possible in all disciplines. Many of the identified rules and guidelines may be limited in application to a very specific situation, or so general as to be common knowledge among design engineers. As field data accumulates, experts should attempt to abstract design heuristics from this information and incorporate it into the design environment based on the applicable specialty area.

Although the goal should to be to develop an environment which distills design expertise and makes it available to all design engineers, research and development in this area should focus on facilitating access to this information rather than trying to replace the design and specialty engineers. This should not be a problem for the foreseeable future since such systems offer no competition to humans in providing the inventive capability that underlies all but the most routine design problems.

Design Process Database

IPTs need a design process database to record the design effort as it proceeds. The information recorded should include design requirements, functional designs, physical designs, and all alternatives considered as well as the justification for selecting or excluding each. The database should ensure that all these items are related, so thai it is possible to trace the allocation of resources and requirements as more detailed designs are developed. The results of evaluations should be recorded, along with the quality of the results. This database underlies the automated aspects of the environment and the various presentations of the design data available to the user.

As more and more IPTs use such an environment, design engineers will have access to a greater number of completed designs. These desigrs will serve several purposes. For example, a design engineer may solve a design problem by reusing an existing design. A frequent design strategy will be to modify an existing design that comes close to satisfying the current requirements. The database will be a source of alternatives to consider at various points in the design process. As time goes by, the environment will more successfully support the proposed methodology because the alternatives available from the database will more completely span the potential design space. As designs recorded in the database are produced and put into use, field experience will accumulate and be stored in the database for each design. Specialty engineers can use this information to refine techniques for estimating design parameters, and all the engineers can use the field data to provide more accurate estimates as the basis of similarity analyses.

Some detail may disappear from the database as the design progresses, in deference to the physical limits on storage space, but information must be retained to explain why choices were made. This information has many uses. Design engineers can use it later in the design process to understand the probable impact of proposed changes and when working on a future design problem to understand how a similar design problem has been solved in the past. Furthermore, if a design fails to meet expectations, the information in the design database could reveal the source of the expectations, and provide a basis for modifying the design or the evaluation procedures.

System for the Interactive Design and Computer Analysis of Reliability

SIDECAR, the prototype tool developed under this contract, explores one approach to facilitating the use of specialty analyses by a design engineer. SIDECAR provides the design engineer the capability to automatically explore the design space associated with a particular design and to identify possible design changes which might improve the quality of the design, as determined by an evaluation function supplied by the design engineer. Historically, design engineers have had to explore the n-dimensional space associated with a proposed functional design manually. As a consequence, few design alternatives associated with a design problem are evaluated unless a proposed design fails to meet the allocated requirements. SIDECAR was developed to assist the design engineer quickly and efficiently explore the design space associated with a design problem to determine if and how the reliability of a proposed design solution could be improved.

SIDECAR operates in conjunction with an electronic computer-aided design (ECAD) system that provides design capture capabilities, a graphical presentation of computed results, and other capabilities that were specifically added for this effort. The ECAD system was extended to enable the design engineer to easily incorporate reliability structures (e.g., triple-modular redundancy [TMR] and error correcting code added to memory) into a design and to provide methods to compute the reliability of these reliability structures. The design engineer can add a reliability structure to the design by simply selecting a base design component and a reliability structure from a menu. Because both the ECAD system and SIDECAR use an object-oriented design representation, the design engineer can also simultaneously model several different functional decompositions and several different physical designs of each functional decomposition. This capability, in turn, enables the design engineer to quickly compare design alternatives (commands are provided that compute the similarities and differences between modeled design alternatives) and to compose a new alternative by branching-off a modeled design alternative or by combining elements of several design alternatives (commands are provided for "cloning" design alternatives).

SIDECAR also incorporates a set of routines that enable it to autonomously propose and evaluate design alternatives that provide the same functions as the candidate design alternative. The results of the design space exploration are displayed as a set of ranked recommendations to the design engineer, who then determines which, if any, of the recommended changes should be adopted. The evaluation function SIDECAR employs to rank the recommendations of the exploration routine may be any algebraic expression (or set of constraints) of arbitrary complexity formed from any of the reliability, performance, and burden (e.g., cost, area, and cooling and power consumption) design parameters or metrics described in Section V. The design engineer may combine them in any way that is desired. The exploration techniques are described below in the order in which they are performed by SIDECAR.

Design Alternatives Exploration

SIDECAR provides the capability to automatically explore the design space associated with a functional design by searching an electronic database for higher quality modules (an assembly of components) or components which may be substituted for an existing module, set of components, or single component in a functional design hierarchy. For those components or modules determined to be functionally equivalent, it computes and displays predicted changes in the requested metrics and the evaluation function. SIDECAR does not limit the search to substituting individual parts; rather, it also includes design fragments previously identified by the design engineer as performing the same function. If, for example, an ASIC and a set of discrete ICs have been defined to perform the same modem functions, SIDECAR will evaluate the impact of substituting the ICs for the ASIC as part of its design space exploration.

Exploration of Temperature Effects

SIDECAR also provides a function to assist the design engineer in determining where active cooling techniques may be beneficially applied to the candidate design. Upon request, the tool performs a temperature sensitivity analysis for each component and module of a proposed design (based upon a temperature sensitivity relation defined by the design engineer) and identifies those components and modules that have the highest temperature sensitivity. This capability allows the design engineer to quickly determine the effects active cooling and changes in the operating environment will have on the reliability of the design. If the design engineer also defines temperature sensitivity relations for other design parameters (e.g., performance), SIDECAR can be used to assess the impact of temperature changes on these parameters.

Exploration of Other Techniques

There are numerous techniques available to increase the reliability of a design besides upgrading the quality of its parts and reducing its thermal stress levels. Table 3[4] lists several of these techniques. However, determining where these techniques may be cost-effectively applied in a design is usually not obvious, especially to an inexperienced design engineer. Even a table that ranks the parts and subsystems of a design by the hazard rate may not provide the insight needed to cost-effectively meet a reliability requirement. SIDECAR allows the design engineer to easily incorporate several of the techniques listed in Table 3 into a candidate design and provides a marginal benefits exploration routine to help determine whether the incorporation of the techniques is justified based on an evaluation function defined by the design engineer. For example, the

[4] From "Cost Analysis Strategy Assessment" by the Defense Systems Management College (1990).

design engineer may incorporate TMR into a design by adding a TMR specification to a level in the design hierarchy. SIDECAR appropriately modifies the design specification and estimates the change in reliability caused by the incorporation of this techniques and the costs (e.g., in terms of area and power) that would be incurred by this change.

Table 3. Reliability Design Techniques

Category	Technique
Fault avoidance	• Reduce electrical, mechanical, and thermal stress levels • Upgrade component technology, packaging, and/or quality • Increase component integration • Improve environmental control
Fault-tolerant - static	• N-modular redundancy with comparison or voting • Error correction codes (hardware or software) • Interwoven logic • Coded-state machines
Fault-tolerant - dynamic	• Reconfiguration (hardware or software) • Standby sparing • Graceful degradation
Fault detection (to support fault-tolerant techniques)	• Duplication and comparison • Error detection codes (hardware or software) • Consistency and capability checking • Time domain detection

SIDECAR also provides a marginal benefits exploration routine to help the design engineer determine where reliability techniques could improve the overall system reliability. Using this routine, the design engineer can easily determine where changes to the system will provide the greatest increase in system reliability as well as the evaluation function.

The design engineer begins the routine by defining the hazard rate (or any other design parameter that is modeled in SIDECAR) to a specific value. Typically, a hazard rate of zero is used. SIDECAR then artificially sets the hazard rate to that value at each level in the design hierarchy and determines the change in the system reliability and evaluation function as each level is artificially altered. The results of this analysis are displayed to the design engineer in a tabular format. This enables the design engineer to determine the marginal effect that improving each part and subsystem of the design to a target level would have on the overall reliability of the design. The routine also enables the design engineer to determine the marginal effect of forcing a particular module or component to a specific hazard rate. Once the design engineer knows where improving the reliability, or any other characteristic, provides the most benefit, the effect on the evaluation function of incorporating more complex techniques such as TMR can be efficiently investigated.

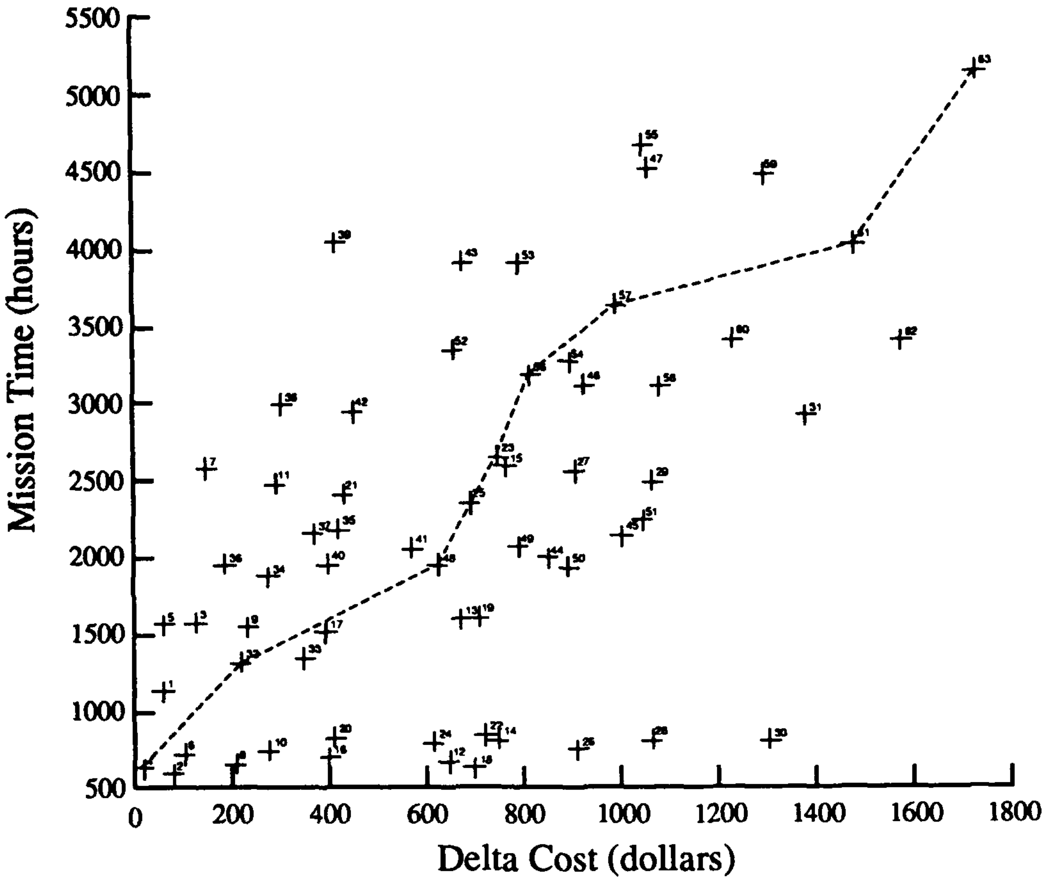
Exploration of Combinational Changes

SIDECAR also provides a general-purpose routine that enables the design engineer to select those component changes, temperature changes, and other modifications that appear to be most beneficial and to determine the effect this set of these changes, and every subset of these changes, would have on the design. For example, if six possible changes were being considered, the design engineer could define the alternatives through the SIDECAR user interface before changing the design. SIDECAR could then be used, through this routine, to analyze the 63 possible combinations of alternatives and provide a graphical or tabular display of the impact each combination would have on various design metrics. One such graphical output is shown in Figure 2. From this graph, the design engineer can quickly determine which combination of changes provides the greatest payoff without exceeding any requirement or budget constraint (e.g., power, cost, and area) and implement that combination of changes.

V. MEASURES OF EFFECTIVENESS

Any approach to design optimization must be based upon an enumeration of the design alternatives and a comparison and evaluation of these alternatives to identify which is superior to the others. To be successful, the comparisons and evaluations must be based on an appropriate set of metrics. This section proposes and briefly describes the most appropriate set of metrics for use by design engineers in comparing and :valuating design alternatives. The list is not intended to be an exhaustive list of all the metrics that should be used to evaluate a design. Instead, it should be considered a list of the primary metrics that design engineers (as opposed to specialty engineers) could use to achieve a locally optimum design solution that they would present to the IPT for evaluation. The proposed optimization approach assumes that the other members of the IPT will employ additional metrics as needed to assess the designs proposed by the design engineers.

This section also includes a discussion of the proper role of each metric and cautions about the misuse of each. Additional information on and formulae for computing some of the metrics are provided in Appendix A.



Note. This figure was copied from a paper written by Yount & Siewiorek (1991). The dotted line represents the path that might be taken by a computerized reliability improvement program that did not first analyze all combinations of alternatives but just took each alternative intc account based on the expected individual increase in the reliabilty metric.

Figure 2
System for the Interactive Design and Computer Analysis of Reliability Sample Output for the Exploration of Combinational Changes

Performance Metrics

There are several basic measures of performance, depending upon whether the intent is to measure the ability to perform a mission or some aspect of the system design (e.g., throughput per unit of time or the precision and signal-to-noise ratio of a device). Except for a measure of mission performance, which is discussed as part of the reliability criteria, there are no other standard performance metrics.

Reliability Metrics

This section describes the recommended reliability metrics. A distinction is made between the metrics to be used in the functional and physical decomposition of the design (i.e., metrics for specification and allocation) and those to be used to enhance the reliability of a proposed candidate design (i.e., metrics for design enhancement).

Metrics for Specification and Allocation

Three basic metrics are proposed for the specification and allocation of reliability: (1) MCSP, (2) mission time (MT), and (3) mean time to failure (MTTF). They differ in the aspect of reliability they address and in their applicability to individual design levels. Each depends on an estimate of the failure rate of the design; thus, the accuracy and reliability of the failure rate directly influence their accuracy and reliability.

Failure Rate. Failure rate is the frequency of failure of an item. For electronics, failure rates are most commonly stated as the number of failures per million hours. The failure rate is rarely used as a design constraint, although it may be used as a goal for component selection or the design of a custom component since it forms the basis for the computation of the other metrics of reliability and, by itself, is only a statistical average.

Mission Completion Success Probability. MCSP is the probability that an item can successfully perform for the duration of the mission for which it has been designed. The mission is described in terms of a set of events or operational states, the duration of each event or state, and the functions required for each event or state. The portion of the mission during which each function is required and the status of that function (i.e., whether or not it is powered up) when it is not needed are part of the mission definition. The loss of a function after it is no longer needed does not affect MCSP as long as the loss of this function does not interfere with any other function that is still needed for the mission. MCSP reflects the benefits of using a redundant or fault-tolerant architecture (e.g., TMR and error correcting codes) to implement a function.

MCSP is frequently specified by the customer because it captures an aspect of reliability that is often a major concern. However, it is difficult to use MCSP as a design requirement at lower levels; it is more appropriately a measure of the performance of a system than the performance of an individual component. The usual approach is to determine the MTTF for each system component that yields the required MCSP. These levels are then allocated to the appropriate components as design requirements.

Mission Time. MT is a measure of the availability of an item—the time during which the reliability of an item is above a specified threshold. The reliability at time *t* is the probability that the system is functional throughout the interval from time *0* to time *t* , given that it was functional at time *0*. Thus, MT is the length of the interval during which the probability of the system functioning without a failure is above a specified threshold. It is related to MCSP but is simpler in concept because it does not take a mission scenario into account. This also means that an MT metric can be computed for an individual component or circuit fragment. Like MCSP, MT reflects the benefits of redundancy because it is unaffected by the failure of redundant elements as long as the remaining elements are sufficient to perform the necessary function.

Mean Time to Failure. MTTF is the average time a circuit or component functions before a failure occurs. Predictions of MTTF do not normally take redundancy or criticality into account. The mean time to critical failure (MTTCF) is the average time a system will operate until a critical failure occurs, assuming the system was initially fully operational. MTTCF reflects the benefits of redundancy, especially for mission critical components.

Mean time between failures (MTBF) and mean time between critical failures (MTBCF) are common, related metrics. Both reflect the average time from the failure of a component or circuit until the system has been returned to service and failed again. They differ from MTTF by taking into account the time required to repair a fault. Thus, they are both reliability and maintainability metrics.

Metrics for Design Enhancement

Although not primary measures of reliability, the metrics described in the next few paragraphs are useful in comparing the level of fault tolerance of design alternatives and in identifying areas of the design that should be explored further.

Inherent Availability. Inherent availability (A_i) is a measure of the fraction of time a system will be operational when logistic delays are neglected (i.e., repair is not dependent upon the availability of a spare part or support equipment). A_i is the ratio of the mean time between maintenance actions (MTBMA) to the sum of MTBMA and the mean time to repair (MTTR). The MTBMA is the mean operating time until system repair or scheduled preventative maintenance, assuming a fault-free system initially. MTBMA is always less than or equal to M'TBCF since it is assumed that a critical system failure will result in a repair action.

Failure Resiliency. Failure resiliency is a measure of the average fault tolerance of a system—the fraction of times a failure in a system will cause a loss of function. It is typically computed as the ratio of MTBCF to MTBF. If the MTTR is much less than the MTBF, the failure resiliency can be approximated by the ratio of MTTCF to MTTF. The failure resiliency of a system is always at least one, and is greater than one as long as there is at least one possible noncritical failure.

Contribution to System Unreliability. To optimize a design, it is useful to determine the fraction of system unreliability (e.g., total number of system failures per unit of time and probability of not completing the mission [1 - MCSP]) that can be attributed to a specific element (i.e., function, module, or component). This metric, when applied to the components of a system, can help identify those elements that contribute most to the unreliability of the system. Recommended metrics of the contribution to system unreliability are failure rate percent and marginal MCSP.

a. *Failure Rate Percent.* This is the fraction of the unit failure rate attributable to a specified function, module, component, or group of functions or components.

b. *Marginal MCSP.* This is a measure of the MCSP when the failures attributable to a level in the design hierarchy (function, module, circuit board, or component) are excluded.

Maintainability Metrics

This section describes the recommended maintainability metrics. It is not appropriate to categorize any of these metrics as more applicable for specification and allocation versus design enhancement. Instead, the metrics may be used in both roles.

The use of three metrics is proposed for maintainability measurement: (1) MTTR, (2) maximum corrective maintenance time (M_{max}), and (3) maintainability index (MI). All three metrics depend on estimating the various types of repairs that may be required by the item being designed, the time each type of repair will require, and the frequency with which it will occur. These estimates take into account the time required to identify and isolate faults, and the effect of ambiguous fault isolation. During design, the calculations are based on the estimated times for a set of maintenance actions that is regarded as a sample from the true population. Although the distribution of maintenance times is clearly not normal (because no maintenance action can take a negative amount of time), the computations are based on the assumption that the sample size is large enough that errors introduced by assuming a normal distribution are negligible. MTTR is the mean of this distribution and M_{max} is a measure of the range of the distribution. MI is an estimate of the total maintenance effort per unit of operating time.

When a system has high reliability goals or constraints, fault-tolerant techniques are often employed in the design. A common fault-tolerant technique is to provision for redundancy. Often, this is combined with scheduled maintenance during which any failed redundant units are identified and replaced to preserve the level of fault-tolerance. In such a situation, scheduled maintenance may be a significant fraction of the total maintenance activity and should be included in the MTTR computation. It is often desirable to have two metrics, one giving the MTTR for faults as they occur, and another including the time for scheduled preventive maintenance. The former is important as an estimate of the likely loss of service during a period of demand; the latter is an important part of determining the total maintenance time required by the system.

Whether preventive maintenance time should also be included in the computation of M_{max} depends on the nature of the system and the customer's point of view. If the system will be used only periodically and the periods of use are less than the interval between preventive maintenance actions, it may be appropriate to leave preventive maintenance out of the M_{max} computation because preventive maintenance can be scheduled for times which do not interfere with periods of system use. On the other hand, if the system is to be essentially in continuous use, the significance of the length of the repair periods is largely independent of the reason for the maintenance.

Mean Time to Repair. MTTR is an estimate of the average time required to repair a fault occurring in the system being designed. This repair time includes the time to identify the faulty item, replace or repair the item, and verify the repair. Any time required to disassemble the system to get at the faulty item and to reassemble the system after the repair is also included in this statistic.

The customer's design requirements often include an MTTR target. This target must be accompanied by a reliability target because a given MTTR can be achieved at the expense of system reliability (items that fail frequently but can be repaired very quickly could be used to achieve the required MTTR level).

Maximum Corrective Maintenance Time. M_{max} is an estimate of the maximum maintenance time required to repair any single fault occurring in the system being designed. M_{max} is usually determined for a specified percentage (typically 90% cr 95%) of all maintenance actions. That is, the intent is for the specified percentage of all maintenance actions to take less than the specified time. M_{max} is required because a design with many short duration maintenance actions and a few very lengthy ones might meet an MTTR requirement but be unacceptable for an application that cannot tolerate lengthy interruptions of service for maintenance. M_{max} gives a goal for the distribution of maintenance times, making it more likely that the result will satisfy customer needs.

Maintainability Index. MI is an estimate of the total maintenance time expended on a system per unit of operating time. This includes likely corrective maintenance actions during a unit of operating time and total preventive maintenance during the calendar time required to accumulate a unit of operating time. For a system in constant use, the operating and calendar times will be the same; however, for a system used periodically, they can differ significantly.

MTTR and M_{max} are measures of the distribution of the length of likely maintenance tasks. By contrast, MI gives an estimate of the total time spent on maintenance tasks and is less sensitive to the length of individual maintenance tasks (except when there are few maintenance tasks or one or more of the tasks have an extremely long duration).

Testability Metrics

The first three subsections of this section (fault detection metrics, fault isolation metrics, and adverse BITE metric) relate to the specification and allocation of requirements. The last section, metrics for design enhancement, relates to evaluation and optimization of a baseline design.

Fault Detection Metrics

Probability of Fault Detection. The probability of fault detection (Pfd) is the most basic and widely used testability metric. It is defined as the conditional probability that a fault will be detected, given that a fault has occurred.

Pfd is an appropriate metric for specification and allocation because it is clearly related to operational costs and is relatively well understood, at least in concept, by design engineers. Unfortunately, the costs of achieving a specified Pfd level may be difficult to predict because Pfd depends on the technology, topology, and operational constraints of the design.

Pfd is usually specified for and allocated to elements of a physical design hierarchy such as a module or box. It may also be specified by device technology such as memory, digital logic, and analog devices. From a maintenance viewpoint, it makes no difference whether one type of device is better-tested than another; however, from a design viewpoint it is much easier and cheaper to test some devices (e.g., memory) than other devices (e.g., analog devices). Underlying the decision to require a certain Pfd value is the belief that greater levels of Pfd will not repay its implementation costs. Since the implementation costs are heavily dependent on the constituent technology, and since the proportions of these within the design are not well-known until a design is well into the detailed design phase, it is often beneficial to specify Pfd by technology. A typical technology-dependent specification is shown in Table 4.

Table 4. Typical Probability of Fault Detection Values

Technology	Pfd
Memory	1.00
Digital ASIC	0.99
Digital Non-ASIC	0.95
Analog	0.90

Specifying Pfd by technology prevents design engineers from ignoring testability in the non-memory portion of a design with large amounts of memory (assuming the required system-level Pfd could possibly be achieved just by testing the memory) and from over-designing tests in difficult-to-test subsystems (it provides an indication of the point of diminishing returns for other technologies).

Undetected Failure Rate. Undetected failure rate (UDFR) is defined as the rate (per unit of time) at which failures that cannot be detected through the designed tests occur. Because it is a poor design parameter (it is highly sensitive to the number of devices in a design, their failure rates, and the tests that will be performed), UDFR is only specified where there is a limit on the number of undetected failures per unit of time that can be tolerated. Pfd should usually be specified instead of UDFR.

False Alarm Rate. The false alarm rate is a measure of false failure indications. It is sometimes expressed as a fraction of total failure indications and sometimes as a number of false alarms per unit of time. False alarms may be caused by transient errors in fault-free hardware or by errors in the diagnostics. Ideally, the false alarm rate would be used to help control characteristics that make a design prone to transients and to persuade design engineers to develop bug-free diagnostics.

In practice, design engineers often try to reduce false alarm rates by such methods as repeating tests and reporting faults only if the retests identify the same faults. The drawback of this practice is that it is extremely difficult to discriminate between a false alarm due to a transient fault and an intermittent failure. A transient fault is due to a temporary environmental condition (e.g., signal interruption or input signal noise) and will occur in a failure-free component. It does not require maintenance since it does not indicate a fault in any component. An intermittent failure is a true fault that requires maintenance but does not manifest itself at all times. For example, a loose connection or a broken wire with the ends still loosely in contact can cause intermittent failures.

Since these types of faults differ only in cause and diagnostics measure effects not causes, any metric that masks transient faults will mask intermittent faults as well. Ensuring that proper diagnostic development methodologies are used and requiring fault insertion and fault-free tests of the diagnostics may be more effective than repeating tests to elicit confirmation as a means of minimizing the number of diagnostic errors.

Cannot Duplicate/Retest OK. Cannot duplicates (CNDs) and retest OKs (RTOKs) are measures of false failure indications, incorrectly diagnosed intermittent errors, and incomplete test coverage. An RTOK event is one in which a fault is initially detected by a test but is undetected by subsequent executions of the same test. A CND event is the failure to detect with test "B" (e.g., a depot test) a fault that was previously detected by test "A" (e.g., a flight-line test) whose coverage was designed to be a subset of test "B."

Because these metrics lump together parameters which design engineers can control (e.g., test coverage) with parameters design engineers have little control over and cannot accurately predict (e.g., false failure indications), they are not useful as design requirements. However, they are useful in-service performance metrics and can provide valuable feedback to specialty engineers. During the design process, it is usually more beneficial to require that depot tests be a superset of the flight-line tests to the maximum extent possible than to require a specific CND/RTOK level.

Fault Isolation Metrics

Five fault isolation metrics are recommended as part of the proposed methodology. These metrics are divided into two types based on the repair and fault containment strategies selected for the design. In cases where a mixture of repair and/or fault containment strategies are planned, a mixture of the two types of isolation metrics should be used.

If a test reveals that one of a set of modules is faulty, two basic strategies may be employed to repair or contain the fault. The first strategy is to replace or quarantine all suspect modules. This "replace all" strategy is effective when the costs of replacing or quarantining all are small compared to the expected cost of not successfully completing the repair or quarantine on the first attempt. The first two metrics described below, percent to n and mean replacement list size (MRLS), should be used for this repair or fault containment strategy.

The second strategy is to replace or quarantine modules one at a time and retest the system after each replacement or quarantine until the fault is repaired. This "prioritized-replacement" strategy is particularly effective if *a priori* probabilities that each module has failed (deduced from detected failure rates) are known so that the items can be replaced or quarantined in the order of decreasing likelihood of failure. The third and fourth metrics described below, percent by n and mean prioritized replacement position (MPRP), should be used for this strategy. The fifth metric described below may be used for either strategy.

Percent to n. Percent to n (e.g., percent to 1, percent to 2, and percent to 3) is a measure of diagnostic resolution or ambiguity. Percent to n is defined as the percentage of faults that will be correctly isolated to *n* or fewer units. This is the isolation metric most often specified in military avionics contracts. It is compatible with the "replace all" strategy only.

When used with a "prioritized-replacement" strategy, the percent to n metric does not accurately reflect the number of units that must be removed and replaced to repair a fault. Consider Examples 1 and 2 of Table 5. Each has a failure that has been isolated by a test to one of two modules; therefore, each example is treated the same by the percent to n statistic (assuming n=2). However, using a "prioritized-replacement" strategy, Example 1 will be repaired with a single replacement 99% of the time, while Example 2 will require two replacements 50% of the time. The percent to n metric treats Examples 2 and 3 differently, but with a "prioritized-replacement" strategy both will be repaired by a single replacement 50% of the time and by two replacements an additional 49% of the time. They differ only in 1% of all failures.

Table 5. Fault Isolation Scenarios

	Percentage of a sample fault caused by each module.		
	Example 1	Example 2	Example 3
Module 1	99%	50%	50%
Module 2	1%	50%	49%
Module 3			1%

This metric is often overemphasized in specifications and allocations. Focusing on i' can yield suboptimal designs since it treats Examples 1 and 2 the same, and distinguishes between Examples 2 and 3. Design engincers may add large amounts of hardware to convert a design like Example 3 to one like Example 2 to reduce the ambiguity group[5] size without any appreciable benefit in terms of the expected number of module replacements.

Mean Replacement List Size. MRLS is the expected number of units that will be replaced to repair a fault, assuming that all suspect units will be replaced. MRLS is compatible only with the "replace all" strategy.

[5] The ambiguity group is the group of components identified by a test or set of tests as containing a fault.

This metric, like percent to n, characterizes the number of suspect modules but allows design engineers more flexibility than percent to n. Using MRLS requirements, design engineers are free to trade, for example, one small list and one large list for twc medium-sized lists without changing the MRLS. In situations where limiting the maximum repair time is important (e.g., limiting turnaround time between missions) percent to n should be used. In situations where the requirement is to minimize average repair time, MRLS should be used.

Percent by n. Percent by n is a measure of fault isolation ambiguity that is compatible with a "prioritized-replacement" strategy. It is defined as the percentage of faults that are expected to be repaired by replacing *n* or fewer modules using a "prioritized-replacement" strategy. The percent to n metric is often inappropriately used in military avionics designs when, in fact, percent by n should be used.

Mean Prioritized Replacement Position. MPRP is a measure of diagnostic ambiguity. It is the expected number of replacements, assuming that units will be replaced one at a time (each time replacing the most likely suspect) using a "prioritized-replacement" strategy. The MPRPs of the examples in Table 5 are $(0.99\times1 + 0.01\times2) = 1.01$, $(0.5\times1 + 0.5\times2) = 1.5$, and $(0.5\times1 + 0.49\times2 + 0.01\times3) = 1.51$ for Examples 1, 2, and 3, respectively.

If repairs are to be accomplished by replacing suspects in a prioritized fashion, and minimizing the number of modules that will be replaced is a primary concern, MPRP is an excellent metric. It is helpful in many situations for which percent to n and other list size metrics are harmful.

As with percent to n and MRLS, the discrete metric, percent by n, should be used when it is desirable to limit the maximum repair, and the expected value MPRP should be used when only the average repair is of concern. Using MPRP places fewer constraints on designs and, therefore, can be expected to result in a better product.

Probability of Correct Isolation. Probability of correct isolation is a measure of error in the diagnostic process. It is defined as the conditional probability that the fault(s) lie within the identified suspect elements, given that fault(s) have been detected and isolation has been attempted. This metric is compatible with both "replace all" and "prioritized-replacement" strategies.

Poorly designed diagnostics and intermittent faults are the main causes of incorrect isolation. Incorrect isolation prolongs the diagnostic process, possibly causing attempted repairs of fault-free units and conclusions that the fault lies within a list of suspects which are actually fault free. This metric is seldom used because these effects can be measured by other isolation metrics such as percent to n. A good reason to use this metric is that, unlike percent to n, probability of correct isolation discriminates between faults that are incorrectly isolated (perhaps due to correctable errors in the diagnostics) and faults that are not isolated. It is most useful if the diagnostics (e.g., model-based diagnostics or intermittent-isolation diagnostics) have a high error rate.

Adverse Build-In Test Equipment Metric

Adverse BITE is a measure of the additional failures introduced by hardware that has been added for test purposes. The BITE fraction is defined as the conditional probability that BITE has failed, given that a hardware failure has occurred. Adverse BITE fraction is defined as the conditional probability that BITE has failed, given that a hardware failure which prevents normal operation has occurred. This excludes many BITE failures which might prevent testing but would not interfere with a mission.

The use of BIT in a design typically requires BITE. BITE generally reduces the performance and increases the basic failure rate of a design, two good reasons for minimizing it. However, using more BITE decreases test development time and complexity, improves fault detection and isolation, and reduces diagnostic errors. Using less BITE generally requires the development of tests which are more complex, thereby increasing the likelihood of costly, error-prone diagnostics.

Metrics for Design Enhancement

Five metrics for design enhancement are described below. The first metric, UDFR, is the only design enhancement metric that is recommended for use as a fault detection metric. The other four metrics are recommended as design enhancement metrics for fault isolation. As with the specification and allocation metrics of fault isolation discussed above, these metrics must be tailored to the replacement strategy (i.e., "replace all," "prioritized-replacement," or a mixture of the two).

Undetected Failure Rate. By estimating the UDFR (discussed above) of modules and sorting the modules by their UDFR, design engineers can get a very clear picture of where the major fault detection problems are. It is recommended that the UDFR be biased by the severity of failures in a fashion similar to a failure modes effects and criticality analysis (FMECA) described in Military Standard 1629A, but using weights derived from Table 1 (Function Criticality) in Section III of this report. Clearly, it is more important to detect some failures than others; the use of weights can help design engineers focus on the most important, rather than the largest, test problems.

Excess Replacement. Excess replacement (ER) is a ıneasure of the number of fault-free modules that would be replaced using a "replace all" strategy. It is defined as the expected number of replacements minus the expected number of faulty modules per unit of time. This metric helps design engineers identify the areas of a design that will have the largest impact on the maintenance effort and the need for spares.

Excess Ambiguity. Excess ambiguity (EA) is a measure of undesirable fault isolation ambiguity (an ambiguity group size of 1 is desired). It is compatible with the "replace all" strategy only. EA is defined as the expected ambiguity group size minus 1.

Excess Prioritized Replacement. Excess prioritized replacement (EPR) is a measure of the number of fault-free modules that would be replaced using a "prioritized-replacement" strategy. It is defined as the expected number of replacements minus the expected number of faulty modules per unit of time. This metric helps design engineers identify the areas of a design that will have the greatest impact on the maintenance effort and the need for spares.

Excess Prioritized Ambiguity. Excess prioritized ambiguity (EPA) is a measure of undesirable fault isolation ambiguity and is compatible with the "prioritized-replacement" strategy. EPA is defined as the MPRP minus 1.

Supportability Metrics

Historically, individual supportability metrics (e.g., mean logistics delay) have not been part of the design specification although the customer often has design constraints and goals related to supportability that affect the design (e.g., limitations on the funds available to procure spares and restrictions on the use of test equipment). As with performance metrics, the metrics that will be used to measure supportability will change with the design problem and be specified by the customer according to the intended usage and deployment of the system. Thus, the three metrics described below should be viewed as examples and not as the only supportability metrics that should be used with the proposed methodology.

Metric for Specification and Allocation

The most important supportability metric is LCC. Key elements of this cost that the design engineer can affect are the numbe. of spares required to support the basic mission assuming no logistics delay, the cost of spares, the need for special support equipment (e.g., ATE and liquid oxygen carts), and special storage requirements (e.g., environmentally controlled storage).

Metrics for Design Enhancement

In addition to the testability metrics ER and EPR, the following metrics are useful for comparing design alternatives for design enhancement for supportability. These metrics have the advantage of allowing designs to be compared on a basis which excludes factors that are outside the design engineer's control (e.g., logistics delays).

Estimated Number of Parts Replaced - "Prioritized-Replacement" Strategy. Estimated parts replacement using a "prioritized-replacement strategy" is the total number of parts likely to be replaced over a given time period. The time period is normally defined by the failure rate such that, if failure rate information is defined for 10^6 hours, this statistic gives the total number of parts likely to be replaced over 10^6 hours.

Estimated Number of Parts Replaced - "Replace All" Strategy. This metric is similar to the previous one except it assumes the "replace all" strategy will be used.

VI. DESIGN TECHNIQUES

Section III describes the basic design process on which the proposed methodology is based. This section describes the RM&S-specific techniques and trades that customize this process for the design of an electronic unit for performance, reliability, testability, maintainability, and supportability. Supportability is not discussed separately to any great degree in this section because it is addressed, for the most part, by the considerations and trades associated with the other aspects of the design.

The basic design process proposed in Section III includes a process of systematically analyzing the applicable requirements at each level in the design hierarchy; investigating alternative approaches to satisfying those requirements via a series of trade studies; and deriving, apportioning, and allocating requirements to the next lower level in the design hierarchy. The process is not a linear progression from a functional to a physical design, but rather a succession of refinements in which alternative implementations, both functional and physical, of a design concept are proposed and evaluated. At each level in the design hierarchy, reliability, maintainability, testability, and supportability trade studies are conducted to establish the characteristics of the design, and a baseline design of the system is selected by assembling the best concepts and alternatives.

The specific trade studies that need to be conducted depend on the mission requirements of the particular system. For military systems, system-level requirements are usually established as a result of trade studies conducted by the DoD and system contractor, and are initially apportioned and allocated to each subsystem based on the results of the trade studies and similarity analyses.

It may be possible, for example, to satisfy mission requirements with a system that is inherently reliable, one that does not need to incorporate fault-tolerant capabilities, or does not employ state-of-the-art components. Alternatively, the mission may require a highly reliable system or one that degrades gracefully in the presence of faults. The system requirements will specify the approach based on the results of the trade studies. For systems with high availability requirements (greater than 0.999), fault-tolerant techniques generally need to be incorporated because fault-avoidance approaches (e.g., using high-reliability components, component burn-in, and careful signal-path routing) will typically not meet the mission availability requirements. High-availability systems often incorporate complex nonserial (redundant) functions and multiple backup modes of operation. Trade studies are normally conducted to determine the best approach to achieving the desired level of availability. Fault-tolerant systems introduce into the design process a level of complexity that systems needing only to be inherently reliable do not. The design of highly fault-tolerant, degradable systems requires a level of sophistication in design knowledge, methods, and tools not needed in the design of systems of lesser complexity.

The basic unit-level RM&S-related design requirements to which the unit is to be designed should be specified by the customer early in the design process (usually during Concept Exploration or Preliminary Design) based on an analysis of mission requirements and the budgetary and organizational constraints of the customer. Although not necessarily optimal, these requirements are assumed to be the best indication of the RM&S levels the design must meet to satisfactorily perform its mission. However, the design requirements should be reviewed and updated as necessary when the mission changes or a technology matures and changes the assumptions upon which these requirements are based.

Performance

The IPT should establish MCSP requirements (or requirements for equivalent measures of mission or system performance) early in the design process. In addition to defining the minimum acceptable MCSP level that satisfies the customer's needs, the IPT should define a maximum MCSP level based on what is believed to be achievable given the customer's schedule and cost constraints. The maximum level is defined in order to provide the IPT with an indication of the difficulty of satisfying the customer's requirement and an idea of the available MCSP design margin. If the margin between the minimum and maximum MCSP is small, the IPT will, in all likelihood, have difficulty meeting the customer's requirement. This same technique may also be employed for many other numerical requirements established by the customer.

Although other performance metrics (e.g., throughput and output signal-to-noise ratio) may be defined by the IPT, these metrics should be viewed as pure performance measurements since they do not reflect the reliability of the system. The following discussion assumes that MCSP is the primary overall measure of system performance since it is affected by the complete range of system factors.

The minimum acceptable MCSP value, the mission definition, and the mission success criteria (i.e., minimum performance and operational levels) are all factors in determining the minimum RM&S a system must achieve. Together they establish a level of readiness each design element must meet during each stage of the mission. This readiness level, in turn, determines the minimum RM&S levels each unit must satisfy. For example, the MTBCF of each unit must be greater than the duration of the mission by an appropriate margin. Similarly, the maintainability (e.g., MTTR and M_{max}) and supportability (e.g., spares availability) of the unit must ensure that the required level of standby readiness can be achieved.

Because the relationship between system characteristics is normally complex (e.g., the MCSP is a function of several factors including mission profile, mission success criteria, system architecture, and the performance and readiness of each subsystem), it is often not possible to solve directly for the RM&S levels which satisfy the mission. The levels, more often than not, must be developed in an iterative fashion in which a preliminary system architecture is proposed, evaluated with the aid of system (and subsystem) reliability models and performance simulations, and modified until the required level of mission performance is achieved. Figure 3 illustrates this process.

Reliability

Several important aspects of reliability need to be considered by the design engineer during the design process. One is a measure of system performance during the mission or the ability of a system to complete the mission for which it is designed. Common metrics for measuring this aspect of reliability include MCSP, MT, MTTCF, and throughput (i.e., the output of the system per unit of time). Another aspect is a measure of how the reliability of a system affects its maintenance requirements, or the probability that the occurrence of a fault will require maintenance actions. Common metrics for this type of reliability are MTTF and MTTCF.

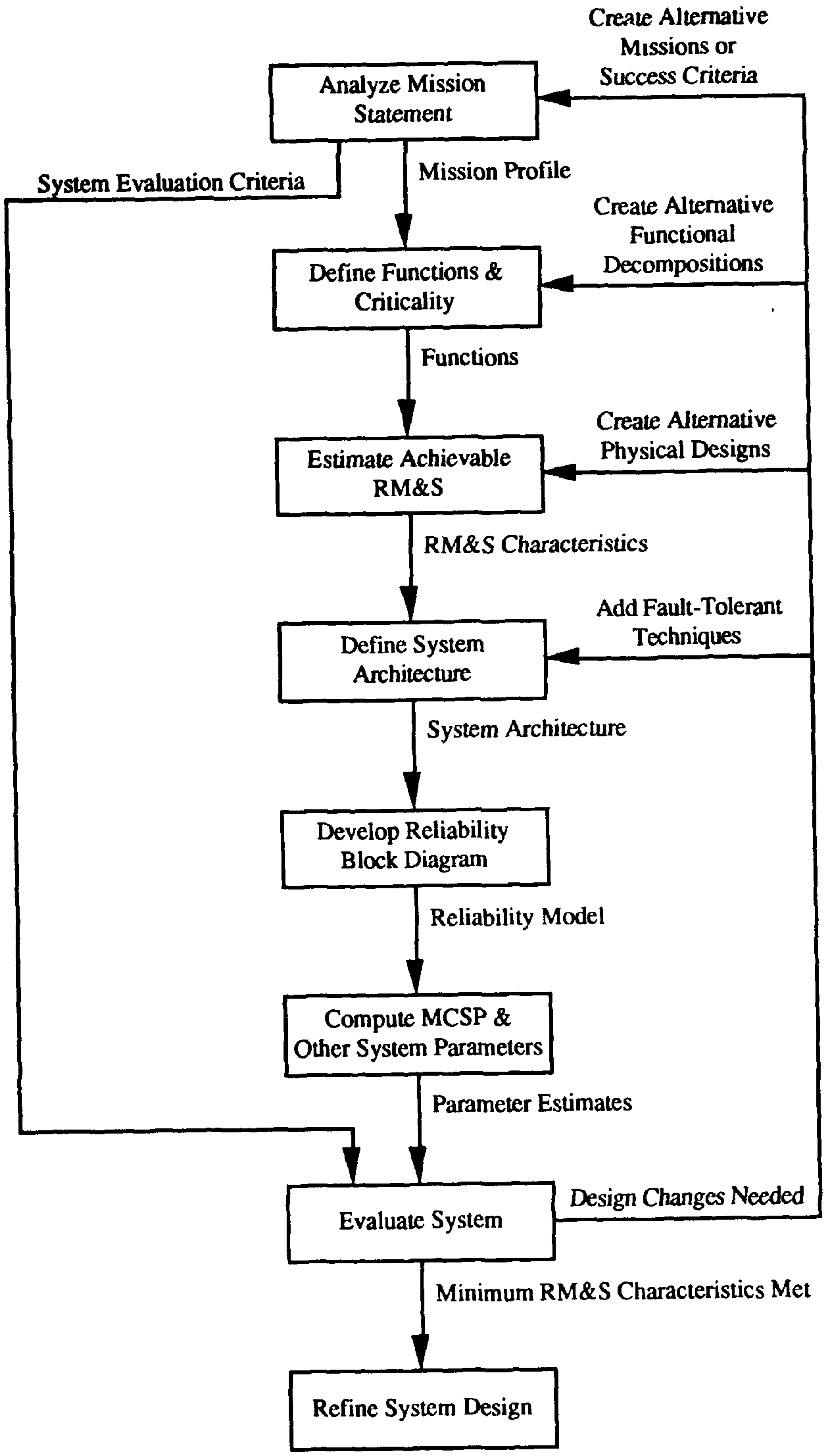

Create Alternative Missions or Success Criteria
Analyze Mission Statement
System Evaluation Criteria
Mission Profile
Create Alternative Functional Decompositions
Define Functions & Criticality
Functions
Create Alternative Physical Designs
Estimate Achievable RM&S
RM&S Characteristics
Add Fault-Tolerant Techniques
Define System Architecture
System Architecture
Develop Reliability Block Diagram
Reliability Model
Compute MCSP & Other System Parameters
Parameter Estimates
Evaluate System
Design Changes Needed
Minimum RM&S Characteristics Met
Refine System Design

Figure 3
System Characteristics Development Process

There is some degree of commonalty in these reliability aspects. In general, the probability that a system will perform satisfactorily during a given mission will increase (an increasing MCSP) as the inherent reliability of a design increases (an increasing MTTF) because decreasing the likelihood of any failure decreases the probability of a failure in a critical function. However, these aspects of the design are sufficiently different that the goal of improving the MCSP of a system may compete with a goal of reducing the maintenance cost because doing so may increase the MTTF or the level of preventative maintenance. For instance, adding redundancy can improve MCSP if the operation of one of two redundant modules is sufficient to perform a critical function, but it will negatively impact MTTF since additional hardware will be added. The addition of hardware increases the probability that some part of the system will fail.

The determination of MCSP is based on a particular mission scenario. Typically, this scenario is a series of events that compose the mission and the operational requirements associated with those events. Some operational requirements will be general and apply to the entire mission while others will be specific to one or more phases of the mission. Each function of the system can be assigned a criticality in the context of such a scenario. Some functions will be required at all times during the mission, some will be required only for certain phases of the mission, some will be recognized as desirable but not essential to the success of the mission, and some will be unnecessary throughout the mission. Criticality is a function of the scenario and the definition of success. For example, if the mission is an attack by a strategic bomber, the phases might be those listed in Table 6 (Scenario 1 refers to a long mission requiring some very low flying and Scenario 2 refers to a short mission with high-altitude bombing).

In this case, the ability to supervise the control surfaces would probably be considered critical for the entire mission. The bombsight would be critical only for events nine and ten (it might need to be tested during prior events). The refueling equipment might be desirable to ensure the plane reaches a friendly base, but not essential to the success of the mission, which requires only events one through ten. If no part of the mission were in range of defensive missile installations, the systems providing protection against attack by such missiles would be unnecessary to the mission. Mission success need not be an all-or-none phenomenon. Certain radars might be critical only for terrain-following flight in events eight through eleven. Without this capability, the mission could continue but the pilot might have to attack an alternate target or have to accept a lower probability of success.

Functions can also be identified as critical because of safety considerations. Safety concerns can be associated with the environment (e.g., minimizing the probability of inadvertently launching a nuclear weapon) or with the crew (e.g., minimizing the probability of losing the avionics system in an aircraft that cannot be flown without the system). Although these safety concerns are part of performance reliability, the design engineer's response to safety concerns is frequently different from his or her response to criticality arising from the mission scenario. This point is discussed in more detail below.

Table 6. Mission Scenarios

	Event	Scenario 1 Functions	Scenario 2 Functions
1	Takeoff		
2	Climb to cruising altitude		
3	Proceed to refueling point	Refueling system	
4	Rendezvous with tanker	Refueling system	
5	Take on fuel	Refueling system	
6	Proceed to start of bombing run	Bombsight test	Bombsight test
7	Descend to terrain-following altitude	Terrain-following radar	
8	Proceed to target	Terrain-following radar, Bombsight test	Bombsight test
9	Acquire target	Terrain-following radar, Bombsight	Bombsight
10	Bomb target	Terrain-following radar, Bombsight	Bombsight
11	Proceed to exit point	Terrain-following radar	
12	Climb to cruising altitude		
13	Proceed to base		
14	Land		

Reliability Design Techniques Versus Burdens

Given that a design is defined by a set of fixed functional requirements, its reliability and availability can seldom be improved beyond that provided by the inherent quality of its components without the use of one or more of the techniques listed in Table 3. This table shows that many potential techniques can be used to achieve a given level of reliability. Because each incurs an associated burden (e.g., design time, monetary cost, weight, and maintenance) and impacts other aspects of the design (e.g., testability), the problem is to select the set of techniques that provides the needed level of reliability while minimizing the total incurred burden.

Minimizing Failure Rate

The fundamental strategy for improving reliability is to minimize the failure rate. Implementing this strategy cost-effectively requires some knowledge of the failure rate for each function. Since functions do not fail (i.e., their physical implementations do), the first step in estimating the failure rate is to propose a physical implementation (or range of possible implementations) for each function. A failure rate can then be estimated based on this physical implementation.

Early in the design process, failure rate estimates are typically based on the failure rate of hardware performing the same or a similar function in an existing system, although they can also be based on the results of research into new technologies or approaches to implementing the function. In both cases, the estimates must be adjusted to account for differences in technology, environment, or functional detail between the hardware on which the estimate is based and the hardware currently being designed. A third source for estimating failure rate data is to estimate the amount of hardware (i.e., number of gates and number of ICs) that would be used to implement the function, and to compute the failure rate from prediction methods such as those in Military Handbook (MIL-HDBK) 217. The accuracy of the failure rate estimates will increase as the design progresses and the amount of hardware can be estimated with more confidence.

Although the use of MIL-HDBK-217 failure rate predictions in the reliability analysis process is under review within the Boeing design community, the general belief is that MIL-HDBK-217-based predictions may be used if they are used in a gross sense—to help identify potential reliability problems and determine the relative (gross) reliability of design elements (on the order of 10 to 20%). The general consensus is that failure rate predictions should be used in conjunction with other indicators of unreliability, such as design rules and derating guideline violations. There is a widely held belief within Boeing that the use of MIL-HDBK-217 failure rate predictions to determine the placement of components on a circuit board for temperature is an inappropriate use of this failure rate data. The approach recommended by the Boeing design community is to select components designed for the temperature environment in which they will operate (ensuring that there is a sufficient temperature "cushion" between the part design temperature and the temperature at which the part is expected to operate) and to ensure that a uniform temperature distribution is maintained across the board during operation (i.e., minimize the number and temperature magnitude of "hot spots").

If the design engineer is considering more than one possible approach to implementing a function, the predicted reliability of the two can be compared. In making these comparisons, the design engineer must identify which functions are critical to mission performance and give the most consideration to the reliability of those functions. Although the critical functions are fixed at the top level on the basis of requirements and mission analyses, the number and nature of critical functions may vary among various possible implementations as the functions are decomposed into subordinate functions. If the difference is significant and the design engineer is confident that the difference is greater than the uncertainty of the estimates (and therefore reflects a real difference in the reliability of the two alternatives), this difference may figure in a trade study to choose between the two approaches. However, reliability is rarely the only basis for choosing between design alternatives and is seldom even the most influential consideration (unless high reliability requirements have been established). Rather, a design engineer should compare many facets of two alternatives, such as performance, production cost, weight, power consumption, risk to the schedule, RM&S, and LCC.

Maximizing Mission Completion Success Probability

To maximize the MCSP, design engineers should use a tool like the Mission Reliability Model (MIREM) (Veatch and Gates, 1986) which derives an estimate of the MCSP based on a mission scenario and the reliability estimates for the various functions. Such a tool can help identify the contribution of each function to the total MCSP so that, if the system requirement is not met, the design engineer has an indication of which functions need to be made more reliable to increase the overall MCSP. The design engineer has several options for improving the reliability of a function. The design engineer could attempt to find an alternative implementation of the function that is more reliable, either by reconsidering options rejected in earlier trade studies or by devoting more effort to identifying other implementations. Also, the design engineer could improve the reliability of the current choice by employing one of the fault-avoidance techniques listed in Table 3 (e.g., upgrading part quality, selecting a more reliable technology, increasing the derating, or increasing the level of integration). A third approach the design engineer could employ is to incorporate one or more of the fault-tolerant techniques listed in Table 3 (e.g., error detection and correction or physical redundancy). Fault-tolerant techniques usually require extra hardware, thereby decreasing the maintenance reliability of the design while increasing the MCSP. One possible approach to incorporating redundancy into a design is to have two hardware elements which perform different, possibly unrelated functions to provide mutual backup. In the event that one element fails, the other could perform the functions of both. MIREM has the ability to model such a design and determine the resulting MCSP.

A recurring decision the design engineer must make during the system design process is the selection of the appropriate level of redundancy. In the early stages of system design, this decision is severely constrained because most levels of the design have not yet been developed. As the design is refined, these decisions must be reevaluated continually to determine whether the redundancy of an item should apply to all components of the item, or whether it should be limited to those components that are critical, important to safety, or particularly likely to fail. In practice, at the avionics system design level this is likely to be limited to determining whether particular instances of redundancy should be implemented at the subsystem, LRU, or board level. In critical situations, redundancy is usually implemented at high levels because an increased number of failure modes and increased opportunities for design errors make redundancy at a lower level riskier. However, redundancy at a lower level may be cheaper because the design engineer can be more selective in the application of redundancy, although more support circuitry (e.g., voting circuits) is likely to be required. Another advantage to redundancy at a lower level is that more faults are likely to be tolerated before the system fails because each redundant group can absorb one or more faults.

A trade study must be performed to establish the best approach to achieving the subsystem availability requirements. As with all trade studies, it should begin with an identification of the assumptions and input data. The study should also establish the equation to be used to determine availability. It should reflect the population of equipments, failure rates of individual components and subsystems, redundancy levels, redundancy management, and repair time or time to restore operations.

Depending on the design requirements, the IPT should investigate several alternative design configurations: no redundancy, dual redundancy, triple redundancy, or a combination of redundancy techniques. The study should be performed as early as possible in the design process for as many alternative configurations as possible. Time constraints and fidelity of the system design concepts dictate how much and to what extent the study can be accomplished. Consideration must be given to such factors as the minimum operational configuration for each subsystem, the time required to switch-in redundancy, the time required to replace nonredundant components, the frequency of failure, and the use of automatic versus manual switch-in of redundant components. The evaluation criteria are generally the same as those used to evaluate most of the trades: hardware resources required, projected engineering costs, operator requirements, skill-level requirements, required system availability, M_{max}, MTTR, spares costs, and the need for and cost of support equipment. Clearly, much of the data the IPT needs to perform this study must be developed by the appropriate design specialty engineers (e.g., reliability and maintainability engineers).

Safety Considerations

Modifications to the circuit to provide safety margins are usually approached somewhat differently. A common solution is to provide redundancy for all functions so that no single failure can render inoperative a function vital to the safety of the system. However, the real concern is to minimize the probability of failure of any function critical to safety. If all these functions have similar failure rates, providing redundancy for all is a reasonable approach to minimizing the probability of failure. However, if the functions differ widely in failure rate, a more detailed analysis is warranted.

In some cases, the safety provision is to have a human take over a function in the event of a failure. If a function must be performed continuously, it may be necessary to provide sufficient redundancy to allow a human enough time to recognize the failure and assume the function in the event of a failure. For example, the failure of the terrain-following radar controlling the flight control system during a terrain-following maneuver may result in a crash if there is no backup radar system and the pilot cannot react quickly enough to assume control of the aircraft and avoid any obstacles.

To properly model safety considerations, the IPT must model failure rates, allowable times for initiation of redundant functions, and achievable times for transfers to various potential redundant functions. In such a case, a human would be modeled as having a probability of failure equal to zero. With such a tool, the responses to safety concerns could be modeled more completely and the available resources could be optimally utilized to minimize the probability of failures compromising safety. One possible drawback to this approach is its dependence on the validity of the assumptions underlying reliability modeling. In particular, reliability modeling depends heavily on the assumption that failure rate is constant over time. If the failure rate varies significantly with time, the analysis will be inaccurate and the design may unnecessarily risk failures that compromise safety.

Allocations

Once a design satisfying the requirements has been developed at one level in the functional design hierarchy, the reliability estimates and redundancy decisions become design requirements for the next lower level. The decision rationale pertaining to the reliability requirements and selected redundancy and fault-tolerant approaches should be documented to facilitate modifications at later levels if it becomes necessary or desirable to reallocate the reliability requirements or modify the redundancy decisions.

Maintainability

The key maintainability design issues are associat ɜd with (1) accessibility to the LRMs; (2) complexity of the unit replacement (including issues such as procedures, weight, and center-of-gravity); (3) use of fault detection and isolation; (4) verification of system operational readiness (e.g., fault-detection logic and circuitry, test vectors, and test sequencing); and (5) preventive maintenance (e.g., unit trimming, calibration, and cleaning).

The two primary maintainability metrics, MTTR and M_{max}, are based on the nature and estimated incidence of the faults that may occur during the service life of a system, and the average time needed to detect, isolate, repair, and verify the repair of each fault.

For a two-level maintenance concept, the field repair task is to identify and replace faulty LRMs. The measures important to this task include the accessibility of each LRM, the Pfd, the resolution of the fault isolation, the difficulty of replacing an LRM, and the equipment needed to verify system status. For modern electronic systems, field preventive maintenance is limited, for the most part, to verifying system operation and unit trimming and calibration in special instances. Decisions which affect these variables also affect many aspects of the system besides maintainability; however, this fact will not be described further in the following discussion. All considerations will be discussed solely from a maintainability point of view.

Accessibility

The physical location, available volume, and shape of the volume are typically allocated physical design constraints. Usually the design engineer only has control over how the electronics are distributed within the allocated volume and the functions that will be assigned to each LRM. To minimize the average repair time, the design engineer (usually the physical design engineer) should distribute the electronics within the volume so that the LRMs most likely to fail are the easiest to replace, sufficient area is provided around each LRM to enable it to be tested without being removed, and LRMs and components within the same test ambiguity group are located together.

Although accessibility does affect the maintainability metrics of an LRM, the time and resources needed to perform fault isolation and testing on an LRM typically have a more significant impact on the maintainability metrics. Consequently, the contribution of the equipment access time to the system metrics can be addressed by a rough-order-of-magnitude estimate (±15 minute) of the time needed for removal and replacement.

Unit Replacement

Unit replacement is simplified by making each unit light, and by minimizing the tools and number of operations required to remove or install a unit. It is further simplified if the connectors are designed to minimize the likelihood of damage from bending pins, over-stressing fasteners, etc., during normal maintenance. Analysis of these requirements will bear on the choice and number of connectors, and on provisions for securing the unit in its mounting. These decisions also influence accessibility, since the difficulty of replacing a unit will be markedly increased if the provisions for securing it necessitate the use of a tool in an area with poor access or minimum clearance. To minimize the average repair time, the electronics should be distributed within the available space such that the LRMs most likely to fail are the easiest to replace; the weight and center-of-gravity of the LRM are within the design guidelines; standard fasteners, etc. are employed; the need for special tools or equipment is minimized; and modules within the same ambiguity group are located together.

Maintenance time is also reduced if the same procedure is used to remove all units. This minimizes the procedures the maintenance technician must learn and the choices that must be made for any single repair. The design engineer can ensure a common procedure by imposing requirements on the design which specify the type of connectors that will be used and how the unit will be secured in its mounting.

The design engineer must place limits on the uniformity of each unit. Provisions should be made to ensure that the unit can only be installed in the correct position and orientation. Similar provisions should be created to ensure that the unit can only be attached to the correct connector. This can be provided by a physical keying arrangement that is independent of the connectors. The design engineer must strike a balance between commonalty of parts and operations and uniqueness of physical fit.

Fault Detection and Fault Isolation

The system engineer must ensure that any of the techniques incorporated in the design to detect and isolate faults are compatible across all subsystems. If unit design requirements do not specify a choice between BIT and external test equipment, the system engineer must address the issue and provide requirements for the detailed design engineer. Effective BIT should reduce maintenance time by providing improved fault-detection and fault-isolation capabilities and by relieving the technician of the need to procure test equipment. Fault identification is likely to be further accelerated because BIT will most likely perform a complete suite of tests faster than would be possible with external test equipment. If BIT is active whenever the system is in use and is connected to nonvolatile storage, it provides the advantage of identifying and documenting a fault when it occurs for subsequent repair. This is especially helpful for intermittent faults which may not be reproducible during troubleshooting and in complex circuits. Under these conditions, it may be difficult for the technician to determine or replicate the conditions under which the fault occurred. BIT also makes it easier to verify and debug a design, because it gives the design engineer more insight into the nature and location of faults that may be observed. The system engineer must balance the advantages of BIT against the increases in failure rate, weight, area, and power dissipation caused by BITE.

Fault detection and fault isolation are also affected by the assignment of functions to hardware design elements during the design process. Fault detection and isolation are simplified when an entire function can be assigned to a single hardware design element. If a single element can provide more than one function, the functions should be related. This approach minimizes the candidates to be considered in the event of any single fault.

A major maintainability decision made at the system level is whether a unit should be designed to be repaired or discarded when it fails. The test hardware and test development effort required to isolate a fault to a replaceable unit is less than that needed to isolate a fault to a replaceable part within a unit. This decision must be based on a comparison of the estimated costs of providing the fault-isolation and repair capabilities with the estimated cost of replacing the unit each time it fails. The costs of providing the isolation include test hardware design and verification, test design, and test hardware production and maintenance. The cost of discarding the unit is based on design, verification, and production costs without the ability to isolate faults lower than the unit level. The decision is heavily influenced by the expected failure rate of the equipment, since most of the costs of fault isolation can be amortized over all failures but the costs of discarding faulty units increase approximately linearly with the number of failures.

A trade study should be used to establish the types of fault-isolation methods that will be used and to select external test equipment. The study should consider a variety of test strategies including self-test on-line (operator initiated), self-test off-line, procedural, use of general purpose test equipment, use of special purpose test equipment, sequenced substitution, and visual inspection. Each strategy should be weighed against alternative designs and equipment types. The overriding factor in selecting a strategy should be cost and the decision should be based on comparisons of testability effectiveness, impact on availability, indirect cost (i.e., training time, spares costs, and personnel costs), and overall complexity.

In conjunction with a status-monitoring trade study (see below), the design engineer should explore the use of alternative fault isolation approaches early in the concept development phase. The results of this study should be a major factor in equipment design. The main inputs to the study should include such factors as fault-isolation overt indications; BIT, BITE, and self-test design, content, and effectiveness; manual test alternatives; use of and requirements for simple, complex, and specialized test equipment; off-line diagnostic routines; and manual intervention/ selective replacement. The evaluation criteria should include equipment design and development costs; system availability requiremer ts; required personnel manning levels, skill levels, training, and experience; M_{max}; MTTR; spares costs; the costs and capability of the installed redundancy; support equipment costs including the cost of the data-monitoring equipment; technical manual costs; and facilities costs.

Status Verification

The design engineer must include provisions for verifying that the system is operational after a repair. The considerations here are similar to those listed above under fault identification and isolation; however, the testing must verify that all necessary system functions can be properly carried out. This testing includes not just the health of each unit, but also the health of the system of cooperating units. If BIT provides this verification, it can provide similar verification each time the unit is started up, or possibly whenever it is requested by the operator. If the verification is to be provided by external test equipment, access must be available without removing any unit. Once again, the necessary provisions will be made by the detailed design engineer but will be directed by requirements established by the system engineer and maintainability engineers.

A status-monitoring trade study should be performed to determine the type and extent of the status monitoring to be employed. For each equipment type within the unit, three types of monitoring should be investigated: no status monitoring, operable status monitoring, and operable status monitoring with incipient failure detection. Depending on time constraints and the availability and fidelity of the information associated with alternative designs, this study can be conducted as early as possible in the design cycle and applied to as many design alternatives as necessary. (Typically, it is performed late in Demonstration/Validation or early in Full Scale Development.)

Preventive Maintenance

As noted in the introduction to this section, the field preventive maintenance tasks are limited to verifying system operational readiness and, in some instances, trimming or calibrating special equipment. For the most part, these situations are obvious to the design engineer because they occur only in special situations such as when no status monitoring has been incorporated into the design.

Testability

The design requirements established by the IPT should include testability goals and constraints. The system engineer should consider using external test, BIT, or both to meet these requirements. If BIT is chosen, the design engineer will need to use existing hardware to provide the test to the extent possible. Where necessary, additional hardware will need to be added to support the test function. These choices will be based on evaluating a set of proposed testability approaches, selecting the best, and revising it as necessary. A basic design strategy is to minimize the testability budget—the resources (e.g., power, board area, and cost)—required to achieve a given level of testability.

In general, improving testability decreases system reliability and performance. In theory, there is an optimal level of testability at which the increase in product cost and the decrease in reliability and performance are justified by the improvements in manufacturing test costs, maintenance, and isolation for fault tolerance. This section presents a set of techniques for creating electronic designs to best meet testability requirements.

Propose Candidate Designs

The design engineer is often presented with a set of requirements which prescribe a level of test thought to be optimal for the subsystem being designed. He or she must propose a subsystem design that provides the required level of test for the minimum cost and performance degradation. This effort may lead to a change in system requirements if it becomes apparent that the specified levels are unnecessarily lax or overly stringent.

The test requirements should incorporate fixed numeric goals and constraints (e.g., a Pfd of at least 0.95), relative requirements (e.g., additional test resources when they result in reduced projected LCC), and testability design rules and practices (e.g., use structured test). Many of the requirements should be computed over all possible faults or, at least, for all faults of a set of specified classes (e.g., stuck signal faults). In the latter case, metrics should be computed over a representative mix of faults expected to be observed in service so that the computation takes into account the anticipated frequency of different fault types.

The most common test requirements that the design engineer should attempt to satisfy are listed below.

- *Pfd*, as mentioned earlier, is the conditional probability that a fault will be detected given that a fault has occurred.

- *Isolation requirements* specify how well the faults are to be isolated. An example is the percent of faults that must be isolated to *n* units. If a fault is isolated to *n* units, the technician can identify a group of *n* units that includes the failed unit responsible for the fault when that fault is observed. Usually there are requirements for n=1, 2, and 3 units. In this context, a unit is the minimum replaceable element. For a two-tier maintenance strategy, a unit can be an LRM for flight-line maintenance, while it is normally a component of an LRM for depot maintenance.

- *Time to detect and time to isolate a fault* specify the time required to detect the occurrence of a fault and the time required to isolate the fault to a specified level of assembly. These requirements are usually given in terms of the maximum time required and the expected time averaged over a representative group of faults.

- *Maintenance strategy* specifies the response when a fault is isolated to a group of units. The two most common possibilities are to replace all items in the group at one time or to replace one item at a time until the item(s) responsible for the fault are identified and replaced. Replacing all the items in an ambiguity group at one time increases the probability of a successful repair on the first repair attempt (unless the fault is intermittent or transient) and is likely to reduce maintenance time at the system level because it eliminates the requirement to verify system readiness after each of a series of candidate replacements. On the other hand, it requires more spares at the operational level and increases work at the depot level since more modules are replaced and sent for repair.

- *Availability of the unit for test* specifies whether the functioning of the unit or portions of the unit can be interrupted during normal operation to run tests. If it can, this requirement indicates the upper bound on the time available for testing during normal operations. This may be specified as a bound on the aggregate of test time over a period of operation, on the maximum length of any interruption, or on both.

- *Overhead allowed for BITE* specifies the amount of allocated resources that may be used to implement BITE technology. BITE is hardware added to the circuit to provide fault identification and isolation. If BITE is to be provided, there are often requirements which limit attributes of this added circuitry such as its power consumption, the area it occupies on the board or chip, its weight, and its failure rate.

- *Safety requirements* specify special requirements on circuitry that influence safety. For instance, there may be especially stringent requirements on the time required for fault detection and the Pfd in such circuitry, or BITE may be explicitly required in circuits critical to safety. Safety requirements may also prohibit testing. For example, in the presence of a single fault, exercising portions of fault-tolerant weapon-control hardware could cause release of the weapon.

- *False alarm incidence* indicates the maximum acceptable rate or percentage of false alarms (indications of a fault when none exists). This requirement applies when BIT is included in a design.

The design engineer should consider a variety of approaches to satisfying the testability requirements. One approach is to generate design alternatives that include combinations of the testability traits listed above as well as those that are specified in the design requirements. Additional parameters that the design engineer should consider in generating these alternatives include the following.

- *Level of maintenance*. The current Air Force standard is two-tier maintenance, but one- and three-tier alternatives are often appropriate for other customers. If there is more than one level, each level must be provided with test resources to identify faults and isolate them to the appropriate level. These resources must be consistent so that a fault identified at one level will also be found by a more detailed test at the next level. Depot-levei maintenance requires extensive inventories, sophisticated test facilities, and highly trained personnel. To reduce LCC and protect critical equipment, these facilities are usually centralized in secure zones. As a result, the design engineer may assume that distributed deployment or deployment to a hostile environment will require multiple maintenance levels.

- *Type and level of te*ʹ*st*. A fundamental test choice is whether to use internal tests using BITE or external tests using ATE. When BITE is chosen, the design engineer may also have to choose between on-line and off-line testing and fault isolation if this choice was not specified in the requirements. The choice between BITE and ATE must be repeated at each maintenance level. BITE requires adding circuitry to the system under design; this increases the cost, weight, area, power consumption, and failure rate. At the same time, it may simplify maintenance by giving an immediaᵃte indication of the source of a fault. If ATE is chosen, the equipment and the tests must be designed and produced, and each technician must have access to the test hardware when performing maintenance.

- *Degree of modularity.* The design engineer must choose between a few large modules or many smaller ones. With many modules, it is more likely that circuitry that is very hard to test can be concentrated on a single module, which can then be discarded rather than repaired when it fails. This can eliminate some low-level test requirements. However, increased modularity may increase the potential for intermittent faults since connectors and interconnects are a major source of these types of faults. Decreasing the number of modules makes it easier to isolate a fault to a module because the entire group of candidates can be located on the same module. This results in a higher probability of successful repair on the first attempt and a reduced incidence of modules arriving at the next maintenance level with no identifiable fault. The disadvantage of this approach is that each of these modules is likely to be more expensive and, therefore, not discardable. Sophisticated equipment is often required to effectively test such modules and the sophistication of the equipment forces the tests to be performed at the depot.

- *Failure mode model.* If it is not specified in the requirements, the design engineer must decide which types of faults will be recognized and recorded by BITE versus ATE. Perhaps the simplest choice is to limit testing to stuck-pin faults. In this model, the only faults are assumed to be those for which an input or output takes on a constant value. More complex alternatives involve such things as identifying intermittent faults, which occur only under some circumstances (e.g., vibration or at certain temperatures) and are independent of the logic values of the circuit, and identifying transient faults, which occur for a short time in response to a temporary condition in the circuit or environment. As noted previously, several major sources of failures and faults are difficult to model.

- *Diagnostic fault tolerance.* The diagnostic fault tolerance parameter determines how many faults can occur without reducing the accuracy of the diagnostics or whether the diagnostic hardware itself is fault tolerant. The complexity of the diagnostics rises quickly as this parameter is increased.

- *Probability of correct isolation.* The probability of correct isolation parameter is the probability that, once a fault has occurred, the diagnostics will include the faulty component in the identified ambiguity group.

- *Additional fault isolation statistics.* If the preceding statistics are insufficient to enable the design engineer to identify the optimal design, additional statistics should be computed to further characterize the adequacy of a test design. If the maintenance strategy is to replace all members of an ambiguity group, the MRLS should be computed for each design alternative.

If the maintenance strategy is to use prioritized-replacement, the MPRP should be used to distinguish between the remaining design alternatives.

- *Test modes.* The test modes parameter determines which functions will be served by test in the design. The design engineer will choose one or more of the following: off-line testing for minimum functionality to support a mission, off-line diagnostics for fault isolation, on-line monitoring of the status of the system, and diagnostics to identify faults as they occur as part of providing fault tolerance.

Evaluate Design Alternatives

Having identified a series of design alternatives, the design engineer must evaluate them to provide a basis for choosing among them. Some alternatives might be eliminated immediately based upon a comparison of their primary evaluation metrics. The remainder should be analyzed in more detail to determine the cost of the testability provisions. The best candidate should be selected based on these analyses.

In addition to the parameters listed above, there are several design parameters that are not part of the testability provisions of the design but constrain the testability decisions including circuit types and the component technology employed in the design. Circuit types include the speed of the circuit, whether it is analog or digital, whether it has a microprocessor, whether it includes memory, and whether it is composed of commercial parts or includes ASICs. These determine the resources potentially available for test and the types of tests required. Possible component technologies include complementary metal-oxide semiconductor, emitter-coupled logic, TTL, and GaAs. These influence the speed of the circuit, the types of faults that can be expected, and the types of resources that will be available for producing diagnostics.

On the basis of the requirements and the choices made for the remaining parameters, a test strategy is selected. Alternatively, the design engineer may investigate a range of test strategies for each candidate design. The test strategy is characterized by the following three parameters.

- The test provisions may be structured (i.e., a predetermined set of tests is provided for each type of component) or unstructured (i.e., diagnostics are provided for one fault or class of faults at a time until the testability requirements have been met). The choice of what to test is determined by the availability of test resources and the difficulty of fault detection.

- Tests may be centralized (i.e., a single aggregation of test resources is used to perform all diagnostics) or decentralized (i.e., some test resources are provided with each block of circuitry to perform local diagnostics).

- The test approach may be stimulation and observation (i.e., a predetermined pattern of inputs is applied to a circuit and the output is compared to expected values) or nonintrusive monitoring (i.e., circuit levels are followed without interfering with the function of the circuit, and each observed value is compared to an expected range or value).

A variety of cost metrics can be computed for each candidate design and test strategy. These include the cost and development time for test development; the overhead for BITE in terms of weight, power consumption, area, performance degradation, and failure rate; the demand placed on system resources by test provisions; and the probability that diagnostics will cause a failure due to a bug or unexpected condition. Some influences of the various design parameters on these cost metrics are indicated in Appendix B.

The overall design requirements will determine which of these is most important. Selection among the candidates can be based on the values of these parameters and the overall goals of the design.

Develop a High-Level Test Plan

Once a design has been selected, the testability decisions are confirmed and documented in a high-level test plan. The test plan describes how a system or subsystem is to be tested. It includes apportionment and allocation of test resources to subsystems, the enumeration of test sets and their purpose, the definition of quarantine requirements (fault containment), and the definition of error-reporting and error-logging procedures.

A test plan serves several functions. First, it confirms that the allocated test requirements can be met by presenting a plan for meeting them. Second, it defines the tests for a system to such a level that implementation costs can be predicted. Third, it provides the abstract design description on which the detailed design of hardware and software can be based. The outputs of the high-level test plan should include hardware modifications (or recommendations) to improve testability, test development estimates, and test performance estimates.

If the system depends on BIT, the design engineer should take maximum advantage of the operational resources. Additional resources should be added with caution, particularly in situations such as military avionics which are very sensitive to increases in weight, power utilization, space requirements, and failure rate. The proposed method of test plan development relies on identifying potential test resources, constraints on their use, and special needs of and constraints on the elements to be tested.

Catalog the Potential Test Resources. The first step in developing a test plan is to identify the resources in each block that can be used to support testing of that block and perhaps other blocks. This identification will be based on an inventory of stimulation, observation, and decision-making resources. If the required resources are not available, they must be created. The three major classes of resources which may be required to support testing are listed below.

- *Data storage resources.* Data storage may be required to store information about error occurrences (including the system state) or to store test patterns and expected results or expected values of various system state parameters. The data storage needs may be met by volatile storage such as dynamic random access memory (RAM), or may require more permanent storage such as static RAM; electrically erasable programmable read-only memory; write once, read many memory; or paper or magnetic media.

- *Output devices.* To be useful, the system must report the test results. If the results are to be reported to a human, these needs can be met by existing displays or other output devices, or by adding displays such as trouble lights. If the results are to be reported to some other part of the system (e.g., an on-board central maintenance system), an interface to another piece of equipment may satisfy the need for an output device.

- *Control and observation resources.* Testing a system or a component usually requires controlling the input to the object under test and observing the response to that input. General purpose processors can provide both control and observation. The use of parity conventions also serves both needs, although in a more limited sense. Event counters, devices to detect transitions between states in a signal, mechanisms to compare several outputs and select the majority value, devices to determine consistency by comparing signals to expected values or ranges, and mechanisms to convert analog signals to digital signals or values are among the alternatives the design engineer should consider to provide signal observability.

Catalog the Topology and Availability of Test Resources. Having identified resources of potential value to system testing, the design engineer must next determine the utility of each for conveying test signals or reporting results while preserving the state of the system under test. The utility is a function of both the availability and the accessibility of the resources. Some resources may not be available as test resources because they are required for a system function and cannot be borrowed (i.e., the system state could not be preserved). Other resources may be available, but topological considerations may render them useless as test resources. This may be due to an inability to control the resource from the outside, the lack of a pathway between the resource and the component to be tested, or the lack of a pathway from the resource to an accessible poir' on the edge of the module.

On the basis of this determination of resources, the design engineer can establish preliminary constraints on the test architecture. In particular, the design engineer can base three preliminary test decisions on the determinations made to date. These decisions are listed below.

- *Select self-test or external test.* If the circuit includes a processor and has good access to internal nodes, the best choice is usually self-test. In other cases, the decision is less clear. The impact of providing missing computational ability or access paths must be weighed against the cost of designing and maintaining external test equipment.

- *Select stimulation and observation paths.* Where possible, operational data paths should be used to limit the additional hardware required for testing. Conflicts with the operational use of these paths or test performance requirements that exceed the capability of operational paths may force inclusion of hardware specifically for test.

- *Select error-reporting paths and protocols.* Error-reporting paths should be short (to make them more reliable). To minimize false error reports and increase the likelihood that observed errors will be accurately reported, the design engineer must make provisions for detecting errors in these paths. Protocols and special hardware are two common alternatives for providing this capability.

Determine What Needs to Be Tested. Once suitable test resources have been identified or provided, the design engineer must decide what to test. The basic approach is to concentrate test resources on the primary-mission functions determined to be the most critical (per the Table 1 ranking), most likely to fail, and the easiest to test. The list of functions to be tested should be expanded until the test requirements (e.g., metrics defining the fault detection and isolation constraints and goals) are met or all of the test resources have been committed. Alternative resource allocations should be explored in those cases where the IPT cannot reach a consensus on the resources that should be allocated to individual functions.

The first step in this process is to perform a FMECA to attribute a cost of failure to each function. Where possible, tests should be specified to detect any faults that affect safety. The remaining faults are ranked according to failure rate and testing is provided for any item that has a significant failure rate and is easy to test. Features that make an item easy to test include self-test capability; small size, hence low complexity; combinatorial logic; redundancy, which makes test by comparison of redundant outputs possible; and pipelining, since a fault in any component in the pipeline can be recognized if the input to the pipeline is controllable and the output from the pipeline is observable.

Tests are provided for the remaining items in order of decreasing failure rate until the testability requirements for the design have been met or the list is exhausted. Items that are particularly difficult to test may be skipped unless their failure rate is so high that requirements cannot be met without testing them. An item may be rendered difficult to test either because of its nature or because of constraints arising from its use in the circuit. Examples of the former include programmable components, for which the test has to reflect the programming and faults can be due to errors in programming or failure of the hardware; high speed circuits, which place stringent timing requirements on the test circuitry; and components which are difficult to control or observe. Examples of constraints arising from the use of a component that make it difficult to test include safety considerations on components like weapons control systems for which the weapon must not be discharged as a result of the test, high throughput requirements which severely limit the time the component is available for test, and limitations on the availability for testing because of particular conditions or mission phases.

Revise as Necessary

If analysis of the design indicates that testability requirements have not been met, the proposed design alternative must be modified to improve testability. This process involves four basic steps: (1) identify and localize major sources of testability problems, (2) propose design alternatives that address these problems, (3) assess the cost and benefits associated with each of the alternatives, and (4) implement a selected subset of design alternatives that appear attractive. This process is based on an analysis of the testability properties of a design. Approaches to performing these analyses are discussed below.

The principal input data for any test analysis are test performance requirements, test performance predictions, and failure modes and their relative frequencies. Ancillary inputs may include predictions of the cost, weight, and area of the proposed design and of the failure rate of required test equipment. These inputs are combined to determine the overall performance of the design and the major sites of poor performance.

Test Performance Requirements. Test performance requirements are usually in the form of fault-detection and fault-isolation goals. These specify the probability that a fault will be detected (given that a fault has occurred), the size of the ambiguity group (if the fault is isolated to a group of elements but not to any one element within the group) to which the fault is isolated when it has been detected, and the time required for fault detection and isolation.

Test Performance Predictions. Test performance predictions are typically predictions of which tests will detect which faults. These predictions may come from manual evaluation of test performance in the presence of each fault identified in a failure modes analysis, simulation of the design in the presence of these faults, analysis of the ability of similar designs to detect and isolate the same type of fault, determination of whether the design violates any of a set of design rules, evaluation of a prototype design in the presence of the faults of interest, or automated analysis of design testability based on models relating design attributes to testability. An example of this automated analysis capability is provided by the Inherent Testability Adviser program developed under this RAMCAD contract and described in a previous report (Tracy et al., 1993).

Some research has shown that fault simulation of low-level functional blocks can produce fault detection and isolation data very close to those derived from detailed, gate-level designs (Defense Systems Management College, 1990); this is not the case for high-level functional blocks such as signal processors and memory management units. Certain design-for-testability methodologies have some highly predictable characteristics (e.g., 100% fault detection for several structured test methodologies) but in general there are many characteristics (e.g., component-level fault isolation or fault latency in the case of structured testing) which cannot be easily or accurately predicted until designs are defined at a detailed level.

Manual testability analyses are inaccurate and extremely labor-intensive. For examp'e, in a recent military avionics design, engineering labor of approximately 1.2 hours per IC was required for manual testability analysis and approximately 12% of pin-level faults (stuck pins) had an incorrect test identified as detecting them.

Simulation-based testability analyses offer the most accurate predictions. Unfortunately, system-level fault simulations require extremely large and powerful computer systems, and even with the largest systems there are limits on how much of the design and what fraction of its operation can be simulated. Furthermore, simulations generally require test stimuli and programs for all programmable devices. These are not usually available until late in the design cycle.

The test requirements, performance predictions, test resource costs, and failure rate predictions are the raw data for analyzing testability of a design. In support of efforts to identify and localize design weaknesses, these data support determination of the contribution of each design component to the overall testability. To evaluate the testability of the design, these contributions are combined to determine the testability statistics of interest for the entire design. The testability metrics presented in Section V, which are primarily fault-detection and fault isolation statistics, form the core of most testability requirements and predictions. This process of combining fault data to identify points of weakness and to evaluate the overall testability of a proposed design is supported by the Statistical Testability Adviser (STA) program designed and prototyped as part of this RAMCAD contract and described in a previous report (Tracy et al., 1993). Since the analysis of testability requires failure rates predictions, STA was developed in conjunction with the SIDECAR program, also developed under the RAMCAD contract (see Section IV). SIDECAR provides STA automated access to the failure rate predictions through an ECAD environment.

These basic testability metrics are often combined with other metrics (e.g., severity of failure or the cost of replacement) to facilitate analyses. To help identify problem areas, the size of the ambiguity group for a given fault can be combined with the cost of the elements likely to be replaced to correct the fault in order to identify points at which reducing the size of the ambiguity group will have the greatest impact on the maintenance cost of the design. To help evaluate proposed solutions to identified problems, the effect of proposed additions on system testability can be combined with the cost of those additions to determine the cost-per-unit improvement in testability.

This process of identifying weaknesses, proposing solutions, evaluating proposed solutions, and implementing the most promising solutions is repeated until the testability requirements of the design are met. The analyses help the design engineer choose the most efficient path to improve the testability of the design. Automated tools such as those prototyped under this contract make this analytical approach to the design of testability feasible in realistic design environments.

Supportability

Supportability requirements are normally defined at the system level and apportioned and allocated to the subsystem design level as a result of an intended-use study conducted by supportability engineers. The intended-use study examines the mission requirements imposed by the customer and those implied by the intended use of the system. From the system usage profile developed in this study, all other supportability and logistics concepts are derived. Ideally, supportability engineers should participate in major design brainstorming sessions so that a common approach can be taken to supporting all subsystems and the supportability of alternative design concepts should be evaluated by comparative analysis to determine the effect of each alternative on system cost and availability.

For electronics design at the unit level and below, the major supportability design issues (other than reliability and maintainability) are unit and component cost; special support equipment needs (e.g., testers and maintenance equipment); present and future component availability; and any requirements (e.g., technician skill, environmental control, and support equipment) associated with testing, handling, or storing its spares. Many support decisions are driven by operational and logistic constraints (e.g., budgetary limitations or restrictions on the use of external test equipment) that cannot be controlled by the design engineer. For a particular system design, many of the parameters that determine the LCC of the system are relatively insensitive to the actions of the design engineer.

Consequently, supportability can best be addressed during the design of a unit by ensuring that cost, reliability, maintainability, and testability allocations are met; by adhering to design rules and guidelines such as those listed in Appendix B; and by seeking to use standard design elements and components from an approved preferred parts list. Doing so will ensure that the modularity (i.e., the granularity of the modules of a design) and commonalty of the unit design are maximized.

REFERENCES

Birmingham, B., Brenan, A., Gupta, A., & Siewiorek, D. (1988). MICON: A single board computer synthesis tool. *IEEE Circuits and Devices Magazine, 4*(1), 37-46.

Defense Systems Management College. (1990). *Cost analysis strategy assessment (CASA).* Fort Belvoir, VA: Defense Systems Management College.

Department of Defense, Military Handbook 217E, Reliability Prediction of Electronic Equipment, January 1982.

Department of Defense, Military Standard 1629A, Procedures for Performing a Failure Mode Effects and Criticality Analysis, November 1980.

Kitzmiller, C.T., & Anderson, M.A. (1991). *Electronic Design Process* (AFHRL-TP-90-34, AD-A234 709). Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

Tracy, M.C., Kitzmiller, C.T., & Anderson, M.A. (1993). *RAMCAD Advanced Research Final Report* (In press). Wright-Patterson AFB, OH: Logistics Research Division, Armstrong Laboratory.

Veatch, M., & Gates, R. (1986). *Mission Reliability Model Users Guide* (AFHRL-TR-86-35, AD-A175 235). Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

Yount, C.R., & Siewiorek, D.P. (1991). SIDECAR: Design support for reliability. In *1991 Design Automation Conference Proceedings.* New York, NY: American Society of Mechanical Engineers.

BIBLIOGRAPHY

Babcock, P., Leong, F., & Gai, E. (1987). *On the next generation of reliability analysis tools.* Unpublished manuscript, NASA Contractor Report No. 178380.

CALS Summer Study. (1988). *Integration of R&M into the automated design process.* Unpublished manuscript, Report of the CALS R&M Summer Study on Complex Electronics.

Cralley, W.E., Dierolf, D., & Richter, K.J. (1990). *Computer Support for Conducting Supportability Trade-Offs in a Team Setting* (IDA Paper P-2313). Alexandria, VA: Institute for Defense Analyses.

Department of Defense, Military Standard 472, Maintainability Prediction, 1966.

Department of Defense, Military Standard 1388-1A, Logistic Support Analysis, April 1983.

Edmond, P. (1989). *Automated synthesis of reliable systems.* Unpublished master's thesis, Carnegie-Mellon University, Pittsburgh, PA.

Elkind, S.A. (1982). Reliability and availability techniques. In D.P. Siewiorek & R.S. Swarz (Eds.), *The Theory and Practice of Reliable System Design* (Chapter 3). Bedford, MA: Digital Press.

Elkind, S. A. (1983, May). *Lambda users manual.* Unpublished paper, Carnegie-Mellon University, Pittsburgh, PA.

Fulton, R.E., Pin-Yeh, C., & Richter, K.J. (1989). *Managing Engineering Design Information* (IDA Paper P-2154). Alexandria, VA: Institute for Defense Analyses.

Goldstein, L.H. (1979). Controllability/observability analysis of digital circuits. *IEEE Transactions on Circuits and Systems, CAS-26*(9), 685-693.

Kelly, G.A. (1955). *The psychology of personal constructs.* New York: Norton.

Leonard, C.T., & Pecht M. (1990). How failure prediction methodology affects electronic equipment design. *Quality and Reliability Engineering International, 6*(4).

Siewiorek, D.P., & Swarz, R.S. (1982). *The theory and practice of reliable system design.* Bedford, MA: Digital Press.

Silberman, G.M., & Spillinger. I. (1990). Using functional fault simulation and the difference fault model to estimate implementation fault coverage. *IEEE Transactions on Computer-Aided Design, 9*(12), 1335-1343.

Stephenson, J., & Grason, J. (1976). *A testability measure for register transfer level digital circuits.* Paper presented at the 6th International Symposium of Fault-Tolerant Computing.

Ullman, D.G., Dietterich, T.G., & Stauffer, L.A. (1988). A model of the mechanical design process based on empirical data. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 2*(1), 33-52.

Yager, R. (1978). *Competitiveness and Compensation in Decision Making: A Fuzzy Set Base Interpretation* (Iona College Technical Report RRY 78-14). New Rochelle, NY: Iona College.

Yount, C. (1990, January). *SIDECAR - A system for the interactive design and computer analysis of reliability: Theory and user's guide* (Draft). Unpublished master's thesis, Carnegie-Mellon University, Pittsburgh, PA.

Zeleny, M. (1982). *Multiple criteria decision making.* New York: McGraw-Hill Book Company.

LIST OF ABBREVIATIONS

A_i	Inherent availability
AL/HRG	Armstrong Laboratory, Logistics Research Division
ASIC	Application-specific integrated circuit
ATE	Automated test equipment
BCS	Boeing Computer Services
BIST	Built-in self-test
BIT	Built-in test
BITE	Built-in test equipment
CND	Cannot duplicate
CPU	Central processing unit
CTE	Coefficient of thermal expansion
DIP	Dual inline package
DoD	Department of Defense
EA	Excess ambiguity
ECAD	Electronic computer-aided design
EPA	Excess prioritized ambiguity
EPR	Excess prioritized replacement
ER	Excess replacement
FMECA	Failure modes effects and criticality analysis
GaAs	Gallium arsenide
IC	Integrated circuit
I/O	Input and output
IPT	Integrated product team
LCC	Life-cycle cost
LRM	Line-replaceable module
LRU	Line-replaceable unit
MCSP	Mission completion success probability
MI	Maintainability index
MIL-HDBK	Military Handbook
MIREM	Mission Reliability Model
M_{max}	Maximum corrective maintenance time
MPRP	Mean prioritized replacement position

MRLS	Mean replacement list size
MT	Mission time
MTBCF	Mean time between critical failures
MTBF	Mean time between failures
MTBMA	Mean time between maintenance actions
MTTCF	Mean time to critical failure
MTTF	Mean time to failure
MTTR	Mean time to repair
NMR	N-modular redundancy
Pfd	Probability of fault detection
PWB	Printed wiring board
RAM	Random access memory
RAMCAD	Reliability, Availability, and Maintainability in Computer-Aided Design
RM&S	Reliability, maintainability, and supportability
ROM	Read-only memory
RTOK	Retest OK
SIDECAR	System for the Interactive Design and Computer Analysis of Reliability
SMD	Surface-mounted device
STA	Statistical Testability Adviser
TMR	Triple-modular redundancy
TTL	Transistor-transistor logic
UDFR	Undetected failure rate
UUT	Unit under test
VLSI	Very-large-scale integration
W	Watt

APPENDIX A

RELIABILITY, MAINTAINABILITY, AND TESTABILITY ALGORITHMS

This appendix identifies and defines a set of metrics proposed as the key metrics a design engineer would use to the perform an initial reliability, maintainability, and supportability evaluation of a proposed design. The set is not intended to be an exhaustive list of all the metrics that should be used to evaluate a design. Instead, it should be considered a list of the primary metrics that design engineers (as opposed to specialty engineers) could use to achieve a locally optimum design solution that they would present to the integrated product team (IPT) for evaluation. The proposed optimization approach discussed in the main body of the report assumes that the other members of the IPT will employ additional metrics as needed to assess the designs proposed by the design engineers.

Reliability Algorithms

The algorithms discussed in this section assume that the failure rate is constant over time, with an exponential reliability distribution. The terms "system," "subsystem," and "element" will be used liberally in this appendix. System will be used as the highest level grouping of components and, as such, can denote either part or all of a design (i.e., the entire system being designed or one of its subsystems or modules). Subsystem will be used to define a discrete section of the system under consideration. The term subsystem will not be used unless a larger system is being discussed. Element will be used to denote any or all of the basic components of a system or subsystem (i.e., an element of a subsystem would refer to a set of components that make up that subsystem). In addition, the Greek letter lambda, λ, is commonly used to denote the failure rate in mathematical equations and will be used in this manner throughout this appendix.

Failure Rate

Component Failure Rate. Failure rate data for individual components can be derived from several sources: (1) the methods defined in Military Handbook (MIL-HDBK) 217, (2) methods or reliability data provided by component part manufactures, or (3) failure rate data obtain from fielded systems. In all cases, component reliability specialists should verify that the methods or data provided to an IPT will yield component reliability estimates representative of a fielded component. In addition to the MIL-HDBK-217 stress methods, many component manufacturers provide estimates of the failure rate of individual components based on component characteristics defined for each component type.

System Failure Rate. The system failure rate, λ_{sys}, is the total failure rate of its elements. This rate is computed by summing the failure rates of each element of the system as shown in Equation A-1.

$$\lambda_{sys} = \sum_{i=1}^{n} \lambda_i \tag{A-1}$$

where λ_i is the failure rate of the i^{th} element and there are n elements.

Percent Failure Rate. The percent failure rate, $\%\lambda_i$, is the percentage of the system or subsystem failure rate that is attributable to a particular element. It can be computed using Equation A-2.

$$\%\lambda_i = \frac{\lambda_i}{\sum_{i=1}^{n} \lambda_i} \bullet 100 \tag{A-2}$$

Reliability and Unreliability

The reliability of a system, $R(t)$, is a measure of the probability that the system will be operating (i.e., is not in a failed state) at time t given that it was operating at time 0. Unreliability, $U(t)$, is a measure of the probability that the system will be in a failed state at time t given that it was operating at time 0. The relationship between reliability and unreliability is shown in Equation A-3.

$$U(t) + R(t) = 1 \tag{A-3}$$

Element Reliability. An exponential relation is the most commonly used method to model the probability of failure distribution over time for single electronic elements. The equation used to determine reliability, $R(t)$, at time t for this method is shown as Equation A-4.

$$R(t) = e^{-\lambda t} \tag{A-4}$$

System Reliability. The reliability of a system depends on its topology, the reliability of its elements, and the specific reliability structures and techniques employed (e.g., redundancy, error correction logic, and dynamic reconfiguration). The following reliability equations assume an independence of failures.

a. *Nonredundant system.* When all elements of a design (or region of a design) must be operational for the system to function properly, a series reliability model is employed. The system reliability, $R_{nonredundant}(t)$, is determined using Equation A-5.

$$R_{nonredundant}(t) = \prod_{i=1}^{n} R_i(t) \tag{A-5}$$

where the system has n design elements and the i^{th} element has a reliability of $R_i(t)$ at time t.

b. *Redundant elements.* When elements of a system are redundant, a parallel reliability model is employed to determine the reliability of the subsystem defined by the redundant elements. If only one of the redundant elements in the subsystem must be operating, the reliability for the subsystem, $R_{redundant}(t)$, is determined using Equation A-6.

$$R_{redundant}(t) = 1 - \prod_{i=1}^{n} \left(1 - R_i(t)\right) \tag{A-6}$$

where the subsystem has n redundant elements. The system reliability is computed by combining the reliability of the subsystem with the reliability for the remaining elements using the series model given in Equation A-5.

c. *Standby sparing.* Standby sparing refers to a system in which spare elements are held in a standby mode and "switched-in" when failure of the primary element is detected. To reflect imperfect detection of the primary element failure, the reliability equation for redundant elements is modified as shown in Equation A-7.

$$R_{standby\text{-}sparing}(t) = R_p(t) + P_{fd}\left(1 - R_p(t)\right) \bullet R_s(t) \tag{A-7}$$

where $R_p(t)$ is the reliability of the primary element at time t, P_{fd} is the probability that a failure in the primary element will be detected, and $R_s(t)$ is the reliability of the spare element at time t.

If the primary and standby elements are identical, Equation A-7 can be generalized for one prime element and *n* standby elements as shown in Equation A-8.

$$R_{standby\text{-}sparing}(t) = R_m(t)\sum_{i=0}^{n}\left(P_{fd}^{i} \bullet \left(1 - R_m(t)\right)^{i}\right) \qquad \text{(A-8)}$$

where $R_m(t)$ is the reliability of the primary element at time t, n+1 is the total number of elements in the primary/standby configuration, and P_{fd} is the probability of detecting a fault in an element given that a fault has occurred.

d. *N-modular redundancy with voting.* In n-modular redundancy (NMR) with voting, *n* elements perform the same function. Their outputs are compared and the output value of the majority is taken to be the correct value. The most common form of NMR is triple-modular redundancy (TMR), in which three elements are included in the system and, if at least two give the same output, the output is taken to be correct. The reliability of such a structure is the probability that either all three elements are operational or only one element has failed and two are operational. The reliability equation for TMR is shown in Equation A-9.

$$R_{TMR}(t) = \left(3 \bullet \left(R_m(t)\right)^2 - 2 \bullet \left(R_m(t)\right)^3\right) \bullet R_v(t) \qquad \text{(A-9)}$$

where $R_v(t)$ is the reliability of the voting element at time t.

Mean Time to Failure

Mean time to failure (MTTF) is expressed in terms of reliability according to Equation A-10.

$$MTTF = \int_{i=0}^{\infty} R(t)dt \qquad \text{(A-10)}$$

Element Mean Time to Failure. Using an exponential distribution of the failure rate, where $R(t) = e^{-\lambda t}$, the MTTF of a single element can be determined using Equation A-11.

$$MTTF = \int_{i=0}^{\infty} e^{-\lambda t}dt = \frac{1}{\lambda} \qquad \text{(A-11)}$$

System Mean Time to Failure. As with reliability, the MTTF of a system depends on its topology, the reliability of its elements, and the specific reliability structures and techniques

MT is a better measure of reliability than MTTF for applications which have a minimum lifetime requirement due to restrictions on maintenance or where failures would have immediate catastrophic consequences. MTTF is a better measure for systems which can tolerate brief periods of downtime and are more easily repaired.

MT can be computed by solving the system reliability equation for time. Using the constant failure rate model, where $R(t) = e^{-\lambda t}$, MT is computed using Equation A-16.

$$MT(r) = \frac{-\ln r}{\lambda} \tag{A-16}$$

Mission Completion Success Probability

Mission completion success probability (MCSP) is the probability that a system that is initially fault-free will complete a defined mission without a fault interfering with any function critical to the mission during a period in which that function is required. Three steps are involved in computing this statistic. First, the mission scenario is analyzed to detennine which functions are required in each phase of the mission. Next, for each function, the probability of performance without a fault during the time span it is required is determined. In the absence of repair capability during the mission, the probability of fault-free performance for a function that is active throughout the mission but required only for one phase must be determined over the time period from the start of the mission to the end of the phase in which the function is required. Finally, these probabilities are combined to determine the probability that all required functions will be available without fault whenever they are required during the mission.

Traditionally, system health is represented by the vector X, where the element X_i is equal to 1 if element i is fault-free whenever it is required during the mission and 0 otherwise. Then, for a particular mission, M, the system structure function $\phi_M(X)$ is 1 if the mission M can be supported with system health X and 0 otherwise. MCSP is then the probability that $\phi_M(X) = 1$. If a phased mission has m phases, let X_i^k equal 1 if component i is up at the end of phase k and 0 otherwise, and let ϕ^k be the structure function for phase k. Then, if $Pr\{event\}$ represents the probability of occurrence of the event, MCSP for a mission of m phases is given by Equation A-17.

$$MCSP = \prod_{k=1}^{m} Pr\left\{\phi^k\left(X^k\right) = 1 \mid \phi^h\left(X^h\right) = 1, h = 1, 2, \ldots, k-1\right\} \tag{A-17}$$

That is, MCSP is the product of the conditional probabilities that $\phi^i(X^i) = 1$ given that the structure function was 1 for all phases prior to phase i.

employed. Thus, a general equation for the MTTF of a system cannot be defined. Instead, an MTTF equation is either derived for a specific system configuration or MTTF is computed by deriving an expression for the reliability of the system, substituting it in Equation A-10, and using numerical techniques to perform the indicated integration.

a. *Nonredundant systems.* The MTTF of a nonredundant system (a system in which no redundancy structures or techniques of any kind are utilized) can be readily computed if the exponential failure rate model can be used for all elements in the system. In this case, the MTTF of the system is computed using Equation A-12.

$$MTTF_{nonredundant} = \frac{1}{\sum_{i=1}^{n} \lambda_i} \tag{A-12}$$

b. *Redundant systems.* If a system incorporates redundant features, the general equation for MTTF (Eqn. A-10) can be integrated numerically.

c. *Standby sparing.* The MTTF equation for a system that incorporates standby sparing of identical elements can be simplified to Equation A-13.

$$MTTF_{standby\text{-}sparing} = \frac{1}{\lambda \bullet P_{fd}} \bullet \sum_{i=1}^{n} \frac{P_{fd}^{i}}{i} \tag{A-13}$$

where there are n identical elements, each with a failure rate of λ and P_{fd} is the probability that a fault in an element will be detected.

d. *N-modular redundancy with voting.* The MTTF of an NMR structure can be computed by integrating the reliability equation as shown in Equation A-10.

Mission Time

Mission time (MT) is the period from the start of a mission to the time at which system or element reliability falls below a threshold reliability level, r. The relationship between reliability and MT is shown in Equations A-14 and A-15.

$$R[MT(r)] = r \tag{A-14}$$

$$MT[R(t)] = t \tag{A-15}$$

The required probabilities that the various functions will be fault-free are computed from formulas for reliability, such as those given previously in the Reliability and Unreliability Section.

Testability Algorithms

Fault Detection

Detected Failure Rate for an Element. The overall detected failure rate, λ_{Du}, for an element, u, depends heavily on the number and capabilities of the tests involved. The rate is defined by Equation A-18.

$$\lambda_{Du} = \lambda_u \sum_{t=1}^{m} FR_{Dut} \tag{A-18}$$

where λ_u is the failure rate of the element being tested, t is the test being conducted, m is the number of tests that test element u, and FR_{Dut} is the fraction of the failure rate of element u that is first detected by test t. Note that the definition for FR_{Dut} assumes that any faults found first by one test will not be included as "found" later by another test.

Undetected Failure Rate for an Element. The undetected failure rate, λ_{Uu}, for an element is related to the failure rate and detected failure rate for that element according to Equation A-19.

$$\lambda_{Uu} = \lambda_u - \lambda_{Du} \tag{A-19}$$

Detected Failure Rate for a Test. The detected failure rate for a single test, λ_{Dt}, which checks numerous elements of a design is defined by Equation A-20.

$$\lambda_{Dt} = \sum_{u=1}^{n} \lambda_u \bullet FR_{Dut} \tag{A-20}$$

where n is the number of elements being tested by test t (i.e., isolation group size).

Probability of Fault Detection for a Set of Elements Tested by a Set of Tests. The probability of fault detection metric, *Pfd*, gives the overall probability of fault detection for groups of elements. The overall probability of fault detection for subsystems and the system can be determined using this metric. The metric is defined by Equation A-21.

$$Pfd = \frac{\sum_{u=1}^{n} \lambda_u \sum_{t=1}^{m} FR_{Dut}}{\sum_{u=1}^{n} USED_u \bullet \lambda_u} \tag{A-21}$$

where *n* is the number of elements under analysis, *m* is the number of tests being used, and $USED_u$ is the usage percentage for element *u*.

Fault Isolation

Percent to N. Percent to *n*, $\%_n$, measures the capability to isolate all faults in a group of elements to *n* or fewer elements. This metric is computed using Equation A-22.

$$\%_n = \frac{\sum_{t=1}^{m} \lambda_{Dut}}{\sum_{u=1}^{n} USED_u \bullet \lambda_u} \bullet 100 \tag{A-22}$$

where λ_{Dut} is the detected failure rate for element *u* under test *t*.

Mean Replacement List Size. The mean replacement list size metric for a set of tests, $MRLS_t$, measures the number of elements to be replaced to repair a fault, assuming that all suspect elements will be replaced in a "replace all" strategy. This metric is computed using Equation A-23.

$$MRLS_t = \frac{\sum_{t=1}^{m} IG_t \bullet \lambda_{Dt}}{m} \tag{A-23}$$

where IG_t is the isolation group size for test *t*.

Mean Prioritized Replacement Position. The mean prioritized replacement position for a set of elements and tests, $MPRP_{ut}$, measures the average number of elements to be replaced to repair a fault, assuming that each element will be replaced in order using the "prioritized-replacement" strategy. $MPRP_{ut}$ is computed using Equation A-24.

$$MPRP_{ut} = \sum_{t=1}^{m} \sum_{u=1}^{n} \frac{\lambda_u \bullet FR_{Dut} \bullet P_{ut}}{\lambda_{Dut}} \qquad \text{(A-24)}$$

where P_{ut} is the ordinal position of element u in a test fault dictionary and the dictionary is ordered su.h that $\lambda_{Du1} > \lambda_{Du2} > \lambda_{Du3}$.

Excess Ambiguity. The excess ambiguity for a group of elements, u, covered by a set of tests, t, is denoted a^s^ EA_{ut} and measures the number of extra elements that will be replaced in a "replace all" strategy. EA_{ut} is computed using Equation A-25.

$$EA_{ut} = MRLS_t - 1 \qquad \text{(A-25)}$$

Excess Prioritized Ambiguity. The excess prioritized ambiguity for a set of tests, t, is denoted as EPA_{ut} and measures the number of extra elements that will be replaced if a "prioritized-replacement" strategy is used. EPA_{ut} is computed using Equation A-26.

$$EPA_{ut} = MPRP_{ut} - 1 \qquad \text{(A-26)}$$

Estimated Parts Replacement for "Replace All" Strategy. The estimated parts replacement for "replace all" strategy metric, EPR_{ra}, provides the average number of parts that will be replaced in a system or subsystem once there is a failure. This metric is normally used more at the system-engineering level than the detailed-design level. EPR_{ra} is computed using Equation A-27.

$$EPR_{ra} = MRLS_t \bullet \sum_{u=1}^{n} \lambda_u \qquad \text{(A-27)}$$

Maintainability Algorithms

All maintainability metrics included in this appendix are based on the ability to accurately predict the failure rate and repair times for all replaceable elements of a design.

Mean Time to Repair. The mean time to repair (MTTR) a system is estimated from all identified faults that could cause the replacement of a system element, the probability of each fault, and an estimation of the mean repair time for each fault. The mean repair time is estimated from the time required to get any required materials and equipment, the time required for each possible way of detecting the fault and the probability that the fault will be detected in that way, the time required to repair or replace the element, and the time required to verify the repair. MTTR is computed using Equation A-28.

$$MTTR = \frac{\sum_{n=1}^{N} \lambda_n \bullet R_n}{\sum_{n=1}^{N} \lambda_n} \qquad \text{(A-28)}$$

where N is the number of replaceable elements, λ_n is the failure rate of element n, and R_n is the mean repair time of element n.

Maximum Corrective Maintenance Tin.e. The maximum corrective maintenance time for the ϕ percentile, $M_{max}(\phi)$, is the maximum time to correct any single fault among the most easily correctable ϕ percent. The term ϕ is usually defined as 90 to 95 percent for military systems. Three equations (Eqns. A-29, A-30, and A-31) are used to derive this metn .

$$M_{max}(\phi) = e^{(\psi \bullet \sigma + MTTR_{ln})} \qquad \text{(A-29)}$$

$$\sigma = \sqrt{\frac{\sum_{n=1}^{N} (\ln R_n)^2 - \frac{\left(\sum_{n=1}^{N} \ln R_n\right)^2}{N}}{N-1}} \qquad \text{(A-30)}$$

$$MTTR_{ln} = \frac{\sum_{n=1}^{N} \ln R_n}{N} \qquad \text{(A-31)}$$

where σ is the standard deviation of the natural logarithms of the repair times, $MTTR_{ln}$ is the mean of the natural logarithms of the repair times, and ψ is the normal deviate corresponding to ϕ. If a variable with normally distributed values is randomly sampled, ϕ percent of the sampled values will be less than $\mu + \psi\phi$, where μ and ψ are the mean and standard deviation, respectively, of the distribution.

Maintainability Index. The maintainability index (MI) is an estimate of the total maintenance time on the system per unit of operating time. It includes both preventive and corrective maintenance. MI is computed using Equation A-32.

$$MI = \frac{t_o \bullet \sum_{n=1}^{N}\left(\lambda_n \bullet M_{c_n}\right) + t_c \bullet \sum_{m=1}^{M}\left(f_m \bullet M_{p_m}\right)}{t_o} \qquad \text{(A-32)}$$

where t_c is a specified period of calendar time over which the metric will be measured, t_o is the total operating time for the system during the period t_c, N is the number of possible faults identified for the system, λ_n is the failure rate of the n^{th} item, M_{c_n} is the corrective maintenance time for fault n, M is the number of preventive maintenance actions for the system, f_m is the frequency of occurrence of preventive maintenance task m, and M_{p_m} is the preventive maintenance time for task m.

APPENDIX B

DESIGN RULES, HEURISTICS, AND GUIDELINES

This appendix contains a partial listing of the design rules, heuristics, and guidelines generally applicable to the design of electronic systems. They are a key element in the overall optimization methodology proposed in this report and are used in two ways: (1) to guide the design engineer in identifying likely design alternatives and (2) to ensure that a design alternative that minimizes the measures of effectiveness does not violate known good design practices.

The design functions and specific design tasks during which each rule, heuristic, or guideline is applicable are indicated by the "Design Function" and "Design Task" fields. The "Affects" field lists the primary design attributes affected by the rule, heuristic, or guideline.

The rules below are loosely organized by function. Each rule is listed with the function to which it is most likely applicable. However, this division is neither firm nor clean. Many rules are applicable in more than one function and many have effects on functions other than the one in which the rule is applied.

General Design Areas

Minimize the complexity of the design by using the simplest possible circuitry to provide the required function. Minimize the number of components and the complexity of each component. The number of functions provided by a component and the number of its failure modes are indicators of the complexity of a component.

Design Function: Electronic Design
Design Task: Circuit Design, Part Selection
Affects: Reliability, Testability, Maintainability

Protect complex, expensive circuitry with cheaper, easier-to-replace components designed to fail at stress levels that will not damage the expensive circuitry.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability, Testability, Supportability

When possible, eliminate the need for components that must be adjusted or aligned.

Design Function:	Electronic Design
Design Task:	Circuit Design, Part Selection
Affects:	Reliability, Testability, Supportability

When possible, limit the number of fan-outs for each internal circuit to N (a fixed number).

Design Function:	Electronic Design
Design Task:	Circuit Design
Affects:	Automated Test Equipment (ATE), Testability, Reliability

When possible, limit the number of fan-outs for each board output to N (a fixed number).

Design Function:	Electronic Design
Design Task:	Circuit Design
Affects:	ATE, Testability, Reliability

Avoid speeds above 5 MHz to minimize signal integrity problems.

Design Function:	Electronic Design
Design Task:	Circuit Design, Part Selection
Affects:	Reliability

Monitor board functionality during thermal-stress testing.

Design Function:	Electronic Design
Design Task:	Circuit Design, Design Verification
Affects:	Reliability, Performance

Circuit Design

Avoid using one-shots.

Design Function:	Electronic Design
Design Task:	Circuit Design
Affects:	Reliability, ATE, Testability

Break up feedback loops by providing connector jumpers and jumper plugs or deactivating circuitry controlled by built-in test (BIT) or ATE.

Design Function:	Electronic Design
Design Task:	Circuit Design
Affects:	Reliability, Testability

Avoid using select-in-test parts. When resistor select kits are used, identify the range of resistors from which to select on the drawing.

Design Function: Electronic Design
Design Task: Circuit Design, Part Selection
Affects: Manufacturing Cost, Repair Cost, Spares Cost

When possible, incorporate current limiters to prevent domino-effect failures.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability, Testability

Do not tie pulled-up or pulled-down do-not-care inputs (e.g., unused complementary metal oxide semiconductor inputs) to inputs that must be in a specific state (e.g., the enable pin on an always-enabled gate).

Design Function: Electronic Design
Design Task: Circuit Design, Schematic Capture
Affects: Reliability, Fault Isolation, Testability

Limit the number of gates tied to a single pull-up or pull-down. Use one 1-KΩ resistor for every ten gates.

Design Function: Electronic Design
Design Task: Circuit Design, Schematic Capture
Affects: Reliability, Fault Isolation, Testability

Part Selection

Establish and use a preferred parts list for each design project. Minimize the number of different part types.

Design Function: Circuit Design, Packaging, Reliability Analysis, Maintainability Analysis, Supportability Analysis
Design Task: Part Selection
Affects: Reliability, Maintainability, Supportability

When possible, use a single logic family. If not possible, use a common signal level for interconnections.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability, ATE, Testability

When possible, use standard connector types.

Design Function: Electronic Design
Design Task: Partitioning, Part Selection
Affects: ATE, Testability

Avoid using select-in-test parts. When selected resistors are used, consider using electrically erasable potentiometers instead (trade parts cost against manufacturing and recalibration cost).

Design Function: Electronic Design
Design Task: Part Selection
Affects: Manufacturing Cost, Maintenance Cost, Spares Cost

When possible, select parts that are well characterized in terms of failure modes; have sufficient available data on the internal structure to support performance, testability, and failure modes analysis; and/or for which simulation models are available.

Design Function: Electronic Design
Design Task: Part Selection
Affects: Performance, Reliability, Testability

When possible, select parts that are independent of refresh requirements. Otherwise, ensure that dynamic devices are supported by sufficient clocking during testing.

Design Function: Electronic Design
Design Task: Part Selection, Circuit Design
Affects: ATE Testability

Ensure that the required part accuracies are within the measurement accuracy of the ATE planned for testing.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability, ATE Testability

Avoid leadless parts on epoxy-glass boards. The chips must be bonded to the board, and mismatches on the coefficients of thermal expansion (CTEs) can cause the bonding or solder joint to crack when the board is heated. Leads provide a cushion to absorb some of the relative motion between the chip and board.

Design Function: Packaging
Design Task: Part Selection
Affects: Reliability

Avoid leadless parts with more than 84 inputs and outputs (I/Os).

Design Function: Electronic Design
Design Task: Circuit Design, Part Selection
Affects: Reliability

If leadless parts are used, specify a ceramic thick-film printed wiring board (PWB) or a polyimide constraining-core PWB.

Design Function: Packaging
Design Task: Part Selection, Mechanical Design of Circuit Board
Affects: Reliability

Avoid leadless parts with more than 44 I/Os on polyimide constraining-core PWBs.

Design Function: Packaging
Design Task: Part Selection
Affects: Reliability

For leadless parts with 45 to 84 I/Os, specify a ceramic thick-film PWB.

Design Function: Packaging
Design Task: Part Selection, Mechanical Design of a Circuit Board
Affects: Reliability

The preferred package for parts with 44 or more I/Os on a surface-mount PWB is a quad pack with gull-wing leads.

Design Function: Packaging
Design Task: Part Selection
Affects: Reliability

Avoid J-leaded parts on boards that cool by conduction through the board.

Design Function: Packaging
Design Task: Part Selection
Affects: Reliability

Avoid leaded or axial tantalum capacitors because solder inside the component may reflow during assembly, causing component failure.

Design Function: Packaging
Design Task: Part Selection
Affects: Reliability, Manufacturing

Dual inline package (DIP) style capacitors are preferred to standup capacitors. Standup capacitors are easily damaged or bent, require a spacer between component and board, have solder fillet problems, and are more susceptible to vibration problems than DIP-style capacitors.

Design Function: Packaging
Design Task: Part Selection
Affects: Reliability, Maintainability

Use sealed connectors to prevent conformal coat contamination of the mating contacts due to wicking.

Design Function: Packaging
Design Task: Part Selection
Affects: Reliability

Avoid connectors requiring potting because thermal expansion of potting material can stress solder joints.

Design Function: Packaging
Design Task: Part Selection
Affects: Reliability, Producibility

Unless standardized, avoid axial parts with power ratings of less than 0.25 watts (W). Such parts have a higher probability of being installed incorrectly because they are hard to identify and orient.

Design Function: Packaging
Design Task: Part Selection
Affects: Reliability, Producibility

Functional Partitioning

When possible, place each function to be tested wholly upon one board.

Design Function: Electronic Design
Design Task: Packaging, Partitioning
Affects: Repair Time, ATE Testability, Maintainability

When more than one function is placed on a board, ensure that each can be tested independently.

Design Function: Electronic Design
Design Task: Partitioning, Circuit Design
Affects: Testability

When possible, partition analog circuits according to their frequency to ease tester compatibility.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Testability

When possible, partition central processing units (CPUs) and their support circuitry from other functions on a PWB.

Design Function: Electronic Design
Design Task: Packaging, Partitioning
Affects: ATE Testability, Physical Partitioning

When possible, place elements of the same ambiguity group in the same package (replaceable item).

Design Function: Electronic Design, Packaging
Design Task: Partitioning, Part Selection
Affects: Fault Isolation, Testability, Repair Time, Repair Cost, Spares Cost

When possible, place pull-up resistors on the same board as the driving components. Place termination resistors as near to the device receiving the signal as possible.

Design Function: Electronic Design
Design Task: Partitioning
Affects: Repair Time, Manual and ATE Testability, Electrical Partitioning

Incorporate blocking gates or tristate devices to allow electrical isolation of functional sections.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability, ATE Testability

When possible, provide separate collector supply voltage and grounds for each functionally independent section.

Design Function: Electronic Design
Design Task: Packaging
Affects: Reliability, ATE Testability

Interface and Connector Design

Establish and use standard connector pin positions for power, ground, clock, test, etc. signals. Stagger ground pins throughout a connector rather than grouping them in one location.

Design Function: System Engineering, Electronic Design
Design Task: Interface Definition
Affects: ATE Testability, Durability

Avoid wire-wrap connections. Such connections have a higher incidence of intermittent faults. Furthermore, intermittent problems are more difficult to resolve because no two items will be identical and manufacturing constraints generally result in ineffective shielding.

Design Function: Packaging
Design Task: Mechanical Design, Routing
Affects: Reliability, Fault Isolation, Testability

Make the number of I/O pins in an edge connector or cable connector compatible with the I/O capabilities of the selected ATE.

Design Function: System Engineering, Electronic Design
Design Task: Interface Definition
Affects: ATE Testability

Arrange connector pins so that the shorting of physically adjacent pins will cause minimum damage.

Design Function: System Engineering, Electronic Design
Design Task: Connector Design
Affects: Reliability, Durability

Use defeatable keying on each board to reduce the number of unique interface adapters required.

Design Function: System Engineering, Electronic Design
Design Task: Interface Definition
Affects: ATE Testability, Supportability

When possible, include power and ground in the I/O connector or test connector. Assess the impact of electromagnetic interference requirements on the power supply.

Design Function: System Engineering, Electronic Design
Design Task: Interface Definition
Affects: ATE Testability, Connector Size, Power Supply Design

Ensure that ground connections contact before power connections to prevent parasitics that can damage components.

Design Function: Packaging
Design Task: Connector Design
Affects: Reliability

Packaging and Mechanical Design

Ensure that the packaging concept is compatible with maintenance and manufacturing concepts. Design electronics to minimize the risk of damaging components during manufacturing.

Design Function: System Engineering, Electronic Design, Packaging, Manufacturing
Design Task: Subsystem Specification, Circuit Design, Mechanical Design
Affects: Reliability, Maintainability, Producibility

Match the CTE of the board and the components or limit the area covered by bonding adhesive to minimize the mechanical stress on solder joints during thermal cycling.

Design Function: Packaging
Design Task: Package Type Selection, Mechanical Design
Affects: Reliability, Manufacturing

When possible, provide for temperature control at sites when power dissipation exceeds 1W. One approach to reducing heat dissipation is to put redundant copies of hot parts in parallel. This is effective only if the voltage can be reduced to maintain a constant total current and if the function of the part can be partitioned among parallel copies.

Design Function: Packaging
Design Task: Environmental Design
Affects: Reliability

Special heat sinks and cooling mechanisms degrade maintainability (e.g., an item may need to be cooled to conduct tests or maintenance) and producibility. Ensure that the cooling scheme is compatible with maintenance and manufacturing concepts.

Design Function: System Engineering
Design Task: Subsystem Specification
Affects: Maintainability, ATE Testability, Manufacturing

Conformal coating may adversely affect reliability, maintainability, or producibility. Ensure that the use of conformal coating is compatible with packaging, maintenance, and manufacturing concepts. Use silicon as the conformal-coating material when surface-mounted devices (SMDs) are used.

Design Function: System Engineering, Packaging
Design Task: Subsystem Specification
Affects: Reliability, Manual Testability, Repair Time, Repair Cost

Consider sealing parts in a dry nitrogen atmosphere to prevent oxidation at connections when high reliability is required.

Design Function: Packaging
Design Task: Mechanical Design
Affects: Reliability, Maintainability, Supportability

When possible, use thermal conductive adhesive to bond parts to boards when increased conductive heat transfer is needed.

Design Function: Electronic Design
Design Task: Circuit Design, Part Selection
Affects: Reliability

If bed-of-nails or other similar test fixtures are to be used, provide test fixture alignment indexes (alignment holes have a 0.125-inch minimum diameter).

Design Function: Packaging
Design Task: Mechanical Design
Affects: ATE Testability, Board Area

If bed-of-nails or other similar test fixtures are to be used, avoid mounting components on both sides of a PWB (bed-of-nails fixtures cannot be used).

Design Function: Packaging
Design Task: Component Placement
Affects: Reliability, ATE Testability, Performance, Board Area

If bed-of-nails test fixtures are to be used, ensure that all component leads protrude sufficiently and uniformly to ensure good contact with test probes, typically 0.040 +/- 0.010 inches.

Design Function: Packaging, Manufacturing
Design Task: Mechanical Design, Lead Trimming
Affects: ATE Testability, Board Area

When SMDs and bed-of-nails fixtures are used, provide test pads or connectors to facilitate probing with a test fixture.

Design Function:	Circuit Design, Packaging
Design Task:	Part Selection, Layout
Affects:	Board Space, Manual and ATE Testability

Use torque-limiting on fasteners to minimize the probability of cracking the board by overtightening.

Design Function:	Packaging
Design Task:	Part Selection, Mechanical Design
Affects:	Reliability, Maintainability

General Maintainability and Testability

A "test earlier because it's cheaper" concept should be considered. Favor low or multiple levels of testing (i.e., test parts, then boards, and then assemblies). Do not create large systems and then try to make them work. Correcting faults detected at higher levels (e.g., system or subsystem levels) is generally more expensive.

Design Function:	System Engineering
Design Task:	Trade Studies, Concept Exploration
Affects:	Testability, Maintainability, Production Cost, Production Test

Assign responsibility for the testability of the design to the design engineer. Test reviews should be part of all internal reviews, the preliminary design review, and the critical design review.

Design Function:	System Engineering, Electronic Design
Design Task:	Partitioning, Circuit Design
Affects:	Testability, Maintainability

The following general mechanical-related maintainability guidelines should be considered.

- Design for minimum usage of screws.
- Use quarter-turn fasteners when possible.
- Make all fuses easily accessible.
- Provide slides on drawer assemblies that also allow the drawer to tilt up and down.
- Provide accessible test points on the front or top of circuits.
- Provide for card extenders.
- Provide easily identifiable cable markings.
- Use quick-connect and quick-disconnect connectors where practical.

- Key all connectors using defeatable keying.
- Identify mating connectors.

  Design Function: Packaging, Electronic Design

  Design Task: Mechanical Design, Part Placement, Detailed Design, Schematic Capture

  Affects: Maintainability, Durability, Testability

The following general software guidelines should be considered.

- Provide the capability to re-initialize, halt, and continue tests.
- Automatically log errors on nonvolatile media, such as paper.
- Provide selected repetition of software blocks, manual override, and a menu option selection.
- Provide help menus, display actual as opposed to expected values, give diagnostic hints, and show messages to the operator.
- At the system level, provide independently executable tests, diagnostics that isolate to the repairable level (if possible), and built-in fault dictionaries.

  Design Function: Test Development

  Design Task: Preliminary Software Design

  Affects: Testability, Maintainability, Repair Time, Repair Quality

When designing analog and hybrid circuits, the following guidelines should be considered.

- Minimize the number of manual adjustments.
- Minimize the use of relays.
- Break feedback loops.
- Provide dividers for high-voltage monitoring points.
- When interfacing signals among cards, boost the signal amplitude to the maximum level possible.
- When using analog test points, monitor the circuit loading, use proper connector types, provide nearby ground terminals, and provide matched impedances and use them for partitioning and visibility.
- Place test points on low impedance points.
- Provide buffers when possible.
- Provide impedance-matching networks.
- Break automatic gain control and automatic frequency control feedback loops.

  Design Function: Electronic Design

  Design Task: Circuit Design, Detailed Design

  Affects: Testability, Reliability

If an external test is expected, ensure that onboard clocks can be overridden and that clock-override interfacing is provided. This allows an inexpensive static tester to be used instead of an expensive dynamic tester, ensuring that the cost of the software and interface adapter is minimized.

Design Function: Electronic Design
Design Task: Schematic Capture
Affects: ATE Testability, Maintainability

Verify that test points do not significantly degrade circuit performance.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Testability, Performance

Buffer and isolate test outputs.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability, Board Area

Provide sufficient built-in self-test (BIST) to isolate faults to a line-replaceable unit.

Design Function: Electronic Design
Design Task: Circuit Design, Part Selection
Affects: Testability, Reliability, Board Area

Provide nonvolatile memory to record BIST findings.

Design Function: Electronic Design
Design Task: Circuit Design, Part Selection
Affects: Reliability, Board Area

Testing should include simulations over the published and expected ranges of performance of the parts used in a design. This information cannot be obtained from laboratory bench testing because, at most, only a few performance-range combinations can be tested and, in most cases, it is not possible to control where a component operates within its published performance-range.

Design Function: Electronic Design
Design Task: Circuit Design, Part Selection
Affects: Reliability

When designing external test connections for microprocessors, ensure that visibility and control points are brought to an operational or test connector. Control points are the data and address buses, clock, tristate control, wait, hold, reset, etc.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: ATE Testability

Do not encapsulate test points that must be accessed during a pre-encapsulation manufacturing test. Keep them accessible for maintenance.

Design Function: Packaging, Manufacturing
Design Task: Mechanical Design
Affects: Manual and ATE Testability, Reliability

BIT and Built-in Test Equipment (BITE) should be used for (or in conjunction with ATE as part of) manufacturing testing to amortize test development cost and reduce the cost and support requirements associated with the required test equipment, to provide an early opportunity to verify functionality of the BIT and BITE, and to ensure that all testing at all sites is performed in a consistent manner with a common set of test resources.

Design Function: System Engineering, Test Development
Design Task: Trade Studies, Initial Test Planning
Affects: ATE Testability, Manufacturing Cost, Manufacturing Lead Time

Provide adequate ground points on the unit under test (UUT) to ensure good grounding when performing diagnostic tests.

Design Function: Electronic Design
Design Task: Schematic Capture
Affects: Manual and ATE Testability, Repair Quality

Partition circuitry in the UUT so that similar circuit classes (e.g., digital, analog, microwave, and video) are on one board. This simplifies test software and reduces both test equipment and adapter costs. This is also true for subclasses of digital circuitry. For example, if a computer is to be split among several cards, keep CPU, memory, and analog interface circuitry on separate cards, thereby allowing a digital function tester, a memory tester, and an analog tester to be used separately.

Design Function: System Engineering, Electronic Design
Design Task: Subsystem Specification, Partitioning
Affects: ATE Testability, Maintainability

Provide initialization capability so that the circuit is placed in a known state when powered up. This saves test software steps and eliminates ambiguities that make the circuit untestable.

Design Function: Electronic Design
Design Task: Schematic Capture
Affects: Reliability, Testability

Consider replacing one-shots with modulo-n counters to allow synchronization of delay elements with test equipment.

Design Function: Electronic Design
Design Task: Schematic Capture, Circuit Design
Affects: BIT, ATE Testability

When cross-coupled gates (switchable latches) are used as a board output, use a buffer to prevent a noise pulse from the interface from being injected into the latch. ATE testability will be improved and reliability may be either improved or hurt depending on the specifics of the design.

Design Function: Electronic Design
Design Task: Detailed Design, Schematic Capture
Affects: ATE Testability, Reliability, Signal Integrity

Insert test points between logic blocks.

Design Function: Electronic Design
Design Task: Detailed Design, Schematic Capture
Affects: Testability, Reliability

As much as possible, group multiple gates from single components (e.g., inverters from a 74FCT04 hexadecimal inverter) in a single functional block and ambiguity group.

Design Function: Electronic Design, Packaging
Design Task: Schematic Capture, Layout, Routing
Affects: Testability, Fault Isolation, Repair Time

Provide test points at logic fan-in and fan-out points. Also, provide test points between digital and analog interfaces.

Design Function: Electronic Design
Design Task: Schematic Capture, Detailed Design
Affects: BIT, ATE Testability

As a general rule, when using large-scale integration or very-large-scale integration (VLSI) devices in the UUT, the following guidelines apply.

- Provide control of clock lines.
- Provide access to address and data buses.
- Provide access to sync or equivalent functions.
- Provide access to hold, reset, and interrupt.
- Provide control over tristate on all devices.
- Provide control of chip select lines.
- Provide access to direct memory access signals.
- Provide access to buffer enable and direction signals.
- Partition analog circuitry.
- Partition microprocessors and their bus integrated circuits from other random combinatorial and sequential logic.

Design Function: Electronic Design
Design Task: Application-Specific Integrated Circuit (ASIC) Design, Detailed Design
Affects: Testability

When designing with surface mount technology, the following guidelines apply for maximum testability.

- Keep the board small.
- When possible, keep all components on one side of the board.
- Use test pads to avoid probing components.
- Keep tall components away from test pads to avoid misalignment of bed-of-nails probes with test pads.
- Provide BIT to help ATE in go/no-go testing.

Design Function: Packaging, Electronic Design, System Engineering
Design Task: Mechanical Design, Part Placement, Schematic Capture
Affects: Testability, Board Space, Durability

Testability

Controllability and Observability

When possible, use active components, such as demultiplexers and shift registers, to enable the tester to control necessary internal nodes using available input pins.

Design Function: Electronic Design

Design Task: Packaging

Affects: ATE Testability

When possible, use active components, such as multiplexers and shift registers, to make necessary internal node test data available to the tester over available output pins.

Design Function: Electronic Design

Design Task: Packaging

Affects: Reliability, ATE Testability

Ensure that redundant elements in the design can be tested independently.

Design Function: Electronic Design

Design Task: Circuit Design

Affects: ATE Testability

When external test equipment will be used to test circuitry, use most of the otherwise unused connector pins to provide test stimulus and control from the tester to internal nodes. A few should be kept open as spares in case additional needs are identified.

Design Function: Electronic Design

Design Task: Circuit Design

Affects: ATE Testability

Ensure that onboard oscillators can be disabled and that all logic can be driven by a tester clock.

Design Function: Electronic Design

Design Task: Circuit Design

Affects: ATE Testability

Break long counter chains into smaller segments to ensure that each segment can be independently controlled by a tester.

Design Function: Electronic Design

Design Task: Circuit Design

Affects: ATE Testability

When possible, provide circuitry to bypass any unavoidable one-shot circuitry.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability, ATE Testability

Ensure that feedback loops can be broken under the control of a tester.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: ATE Testability

In microprocessor-based systems, ensure that the tester has access to the data bus, address bus, and all important control lines.

Design Function: Electronic Design
Design Task: Unit Partitioning, Circuit Design
Affects: Reliability, ATE Testability

Include test control points at those nodes that have high fan-in (i.e., test bottlenecks).

Design Function: Electronic Design
Design Task: Unit Partitioning, Circuit Design
Affects: Reliability, ATE Testability

When possible, provide input buffers for control-point signals with high drive capability requirements.

Design Function: Electronic Design
Design Task: Unit Partitioning, Circuit Design
Affects: Reliability, ATE Testability, Area Consumption

When possible, provide unused connector pins to offer additional internal node data to the tester.

Design Function: Electronic Design
Design Task: Partitioning, Packaging
Affects: Reliability, ATE Testability

Ensure that signal lines and test points are designed to drive the capacitive loading represented by the test equipment.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability, ATE Testability

When possible, provide test points such that the tester can monitor and synchronize to onboard clock circuits.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability, ATE Testability

When possible, place test access points at those nodes that have high fan-out.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability, ATE Testability

When possible, employ buffers when the test point is a latch and is susceptible to reflections.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability, ATE Testability

Buffer test points to prevent the test equipment from damaging internal circuitry. Ensure that each test point is adequately buffered or isolated from the main signal path.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: Reliability

Initialization

Ensure that the circuitry can be quickly and easily driven to a known initial state (e.g., master clear, less than N clocks for initialization sequence).

Design Function: Electronic Design
Design Task: Circuit Design
Affects: ATE Testability

Keep item warm-up time reasonable.

Design Function: System Engineering
Design Task: Subsystem Specification
Affects: Testability

Analog Circuits

When possible, ensure that one test point per discrete active stage is brought out to the connector. Ensure that cascaded stages of operational amplifiers and transistors are isolated or testable.

Design Function:	Electronic Design
Design Task:	Partitioning, Packaging
Affects:	Reliability, ATE Testability

Avoid multiple, interactive adjustments.

Design Function:	Electronic Design
Design Task:	Packaging
Affects:	Maintainability

When possible, partition the design so that the circuits of each UUT are functionally complete (e.g., another UUT is not required to provide the needed bias networks or loads).

Design Function:	Electronic Design
Design Task:	Partitioning
Affects:	ATE Testability

Minimize the number of multiple phase-related or timing-related stimuli.

Design Function:	Electronic Design
Design Task:	Circuit Design
Affects:	ATE Testability

Minimize the number of phase or timing measurements.

Design Function:	Electronic Design
Design Task:	Circuit Design
Affects:	ATE Testability

Minimize the number of complex modulation or unique timing patterns.

Design Function:	Electronic Design
Design Task:	Circuit Design
Affects:	ATE Testability

Ensure that the required test stimulus frequencies, rise times, amplitude, and pulses are compatible with tester capabilities.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: ATE Testability

Ensure that response measurements include frequency, rise time, amplitude, and pulse width measurements that are compatible with tester capabilities.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: ATE Testability

Digital Circuits

Ensure that all clocks of differing phases and frequencies are derived from a single master clock.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: ATE Testability

When possible, ensure that all memory elements are clocked by a derivative of the master clock (avoid gate clocks). Avoid elements clocked by data from other elements.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: ATE Testability

Ensure that all test buses have a default value when selected.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: ATE Testability

Ensure that a known output is defined for every word in a read-only memory (ROM). Ensure that the improper selection of an unused address will result in a well-defined error state.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: ATE Testability

When possible, ensure the design is free of WIRED-ORs.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: ATE Testability

If the design utilizes a structured testability design technique (e.g., scan path and signature analysis), ensure that all design rules are satisfied for that design.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: ATE Testability

Built-In Test

Ensure that BIT in each item can be exercised under the control of the test equipment.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: BIT Effectiveness, ATE Testability

Make onboard BIT indications available at the I/O connector.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: BIT Effectiveness, ATE Testability

When possible, develop BIT using a building block approach (e.g., all inputs to a function are verified before that function is tested).

Design Function: Electronic Design
Design Task: Circuit Design
Affects: BIT Effectiveness, ATE Testability

Ensure that BIT makes maximum use of the mission circuitry. The optimal allocation of BIT makes effective use of hardware, software, and firmware.

Design Function: Electronic Design
Design Task: Circuit Design
Affects: BIT Effectiveness, ATE Testability

When possible, onboard ROM should contain self-test routines.

Design Function:	Electronic Design
Design Task:	Circuit Design
Affects:	BIT Effectiveness, ATE Testability

Processing or filtering of BIT sensor data should be performed, when possible, to minimize BIT false alarms.

Design Function:	Electronic Design
Design Task:	Circuit Design
Affects:	BIT Effectiveness, ATE Testability

When possible, BIT should incorporate techniques to verify faults and errors detected during startup.

Design Function:	Electronic Design
Design Task:	BIT Circuit and Software Design
Affects:	Reliability, Testability, False Alarms

When possible, BIT should use adjustable thresholds for rate, range, and time-interval checks. This provides a mechanism to tune the BIT to reflect the actual operational and maintenance environment. *Overly stringent thresholds are often a source of false alarms.*

Design Function:	Electronic Design
Design Task:	BIT Circuit and Software Design
Affects:	Reliability, Testability, False Alarms

BIT should give information about degree of failure, not just a binary value.

Design Function:	Electronic Design
Design Task:	Circuit Design, Part Selection
Affects:	Reliability, Testability

Tailor the data provided by BIT to the differing needs of the system operator and the system maintainer.

Design Function:	Electronic Design
Design Task:	Circuit Design
Affects:	BIT Effectiveness, ATE Testability

☆U.S. Government Printing Office: 1993 - 750-061/60287