4

RADC-TR-88-69, Vol II (of two)
Final Technical Report
October 1989

R/M/T DESIGN FOR FAULT TOLERANCE, TECHNICAL MANAGER'S DESIGN IMPLEMENTATION GUIDE

Grumman Aerospace

Stanley J. Murn, Jr.

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

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

RADC-TR-88-69, Vol II (of two) has been reviewed and is approved for publication.

APPROVED: Joseph A. Caroli

JOSEPH A. CAROLI
Project Engineer

APPROVED: John J. Bart

JOHN J. BART
Technical Director
Directorate of Reliability & Compatibility

FOR THE COMMANDER: James W Hyde III

JAMES W. HYDE III
Directorate of Plans & Programs

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (RBET) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

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

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
N/A

2a. SECURITY CLASSIFICATION AUTHORITY
N/A

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE
N/A

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
N/A

5. MONITORING ORGANIZATION REPORT NUMBER(S)
RADC-TR-88-69, Vol II (of two)

6a. NAME OF PERFORMING ORGANIZATION
Grumman Aerospace Corp

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Rome Air Development Center (RBET)

6c. ADDRESS *(City, State, and ZIP Code)*
Aircraft Systems Division
S Oyster Bay Rd
Bethpage NY 11714

7b. ADDRESS *(City, State, and ZIP Code)*
Griffiss AFB NY 13441-5700

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Rome Air Development Center

8b. OFFICE SYMBOL *(If applicable)*
RBET

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F30602-85-C-0161

8c. ADDRESS *(City, State, and ZIP Code)*
Griffiss AFB NY 13441-5700

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
62702F	2338	02	2F

11. TITLE *(Include Security Classification)*
R/M/T DESIGN FOR FAULT TOLERANCE, TECHNICAL MANAGER'S DESIGN IMPLEMENTATION GUIDE

12. PERSONAL AUTHOR(S)
Stanley J. Murn, Jr.

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM Oct 85 TO Jul 88

14. DATE OF REPORT *(Year, Month, Day)*
October 1989

15. PAGE COUNT
228

16. SUPPLEMENTARY NOTATION
N/A

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
13	08	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Reliability, Fault Tolerance,
Maintainability, Design Guidance.
Testability,

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Fault tolerance has come into almost universal use in modern day systems of all types. This report contains design guidance and general information for Air Force and contractor technical managers and system integrators with respect to the nature and form of Reliability/Maintainability/Testability (R/M/T) considerations as they apply to fault tolerant systems. This technical managers design implementation guide contains instructions for accomplishing the R/M/T programmatic tasks of MIL-STDS 785, 470 & 2165 for fault tolerant systems development. Important fault tolerance design options and tradeoff analysis methods are discussed to aid the technical manager or system integrator in understanding and managing the entire fault tolerant system design process. This report is Volume II of II. Volume I is an R/M/T Design for Fault Tolerance, Program Manager's Guide. Volume II contains a more in-depth view of the technical issues regarding R/M/T fault tolerance design techniques.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Joseph A. Caroli

22b. TELEPHONE *(Include Area Code)*
(315) 330-4205

22c. OFFICE SYMBOL
RADC (RBET)

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

EXECUTIVE SUMMARY

Fault tolerance plays a key role in the development and successful operation of command, control, communication and intelligence (C^3I) systems. The design and engineering activities during the systems planning and requirements development phases focus on achieving a proper balance of fault tolerance and reliability, maintainability, and testability (R/M/T). It is imperative that the technical manager be certain that the system configuration selected and refined during these phases will achieve all applicable R/M/T requirements. Furthermore, the R/M/T design activities identified in the planning stage must be rigorously pursued during subsequent design and development phases.

The mission capabilities of C^3I systems can be significantly and effectively enhanced by proper application of fault tolerant design techniques. Technical managers must understand the implications of decisions made and must control the configuration selection process to avoid unnecessary complexities that would contribute little to system capability but might increase life cycle cost. System performance, supportability, and the cost of competing fault tolerance approaches must be clearly defined early in the development phase to support critical design configuration decisions that will be made during later phases.

This *Technical Manager's Design Implementation Guide* was prepared by the Aircraft Systems Division of the Grumman Corporation under RADC contract F30602-85-C-0161, entitled *R/M/T Design for Fault Tolerance*. The objective of this document is to provide Air Force and contractor technical managers with guidance on how to implement fault tolerant designs by using state-of-the-art R/M/T fault tolerance techniques. A previously published guide under this contract, entitled *R/M/T Design for Fault Tolerance - Program Manager's Guide* (Ref 32), contains a detailed discussion of R/M/T program planning and management requirements for fault tolerant systems. These Guides were developed to help structure

and tailor cost-effective programs for reliable, maintainable, and testable fault tolerant C^3I systems.

This *Technical Manager's Design Implementation Guide* provides the essential information that Air Force and contractor technical managers need to control fault tolerant C^3I system development by means of specification, design, analysis, and tradeoff. The Guide addresses the following critical areas:

- Fault tolerant design methodology
- R/M/T and software program planning and management
- Specification of fault tolerance and R/M/T requirements
- R/M/T interrelationships and impact on fault tolerant design
- Hardware and software fault tolerant design options
- R/M/T evaluation and tradeoff analyses.

This Guide is intended for use both as a reference document and a tutorial aid. The examples presented illustrate areas of application, potential benefits that can be derived from fault tolerant design features, and their limitations. In addition, checklists are located at the end of major sections to provide the technical manager with a convenient reference of the major R/M/T impact areas and issues to consider in future fault tolerant C^3I development programs.

PREFACE

This Technical Manager's Design Implementation Guide was prepared by the Reliability, Maintainability and Safety Section of the Grumman Aircraft Systems Division, Bethpage, New York, for Rome Air Development Center, Griffiss Air Force Base, New York. Mr. Joseph Caroli (RBET) was the RADC Project Engineer.

This Guide was developed in the period between September 1985 and July 1988. In addition to the author, Stanley Murn, Jr., Grumman study team contributors included Messrs. Gary Bigel, David Conroe, Allan Dantowitz, John DiLeo, John Golden, Theodore Gordan, Kenneth Haller, John Kappler, Victor Pellicione, Frank Perazzo, and George Pflugel.

CONTENTS

CONTENTS (contd)

CONTENTS (contd)

Section Page

CONTENTS (contd)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (contd)

LIST OF ILLUSTRATIONS (contd)

LIST OF ILLUSTRATIONS (contd)

1 - INTRODUCTION

Reliability, maintainability, and testability are essential system attributes required to achieve the demanding C^3I program objective of high system effectiveness within acceptable life cycle cost. Therefore, Air Force and contractor technical managers must understand and control the fault tolerant design process to assure that these attributes are not compromised.

A fault tolerant system design is one with provisions to avoid a system failure after hardware or software faults have occurred within the system. Fault tolerance must be incorporated into a design as part of the system engineering process and directed at critical design areas. The objective of this study is to provide the technical manager with guidance in the design of fault tolerant C^3I systems. Design and programmatic options such as redundancy techniques, reconfiguration strategies, opportunistic maintenance, testing, and monitoring are presented for use by technical managers/system integrators who may not be intimately familiar with these design enhancement techniques.

This Guide identifies the roles of Air Force and contractor technical managers in the fault tolerant system design process. Air Force technical managers must assure that fault tolerant C^3I system performance and R/M/T requirements are met prior to final approval of the design configuration. Prime contractors and systems integration contractors develop and optimize design concepts and configurations that satisfy the system requirements. To assure a cost-effective program, Air Force and contractor technical managers must cooperate to formulate realistic (achievable) system requirements and be familiar with methods used to conduct R/M/T design tradeoff analyses and evaluations. Therefore, the material presented herein is relevant to both Air Force and contractor technical managers. Checklists are provided at the end of several sections of this Guide. These checklist questions are particularly important at the system requirements review (SRR), preliminary design review (PDR) and critical design review (CDR) to supplement the reliability and maintainability evaluation criteria listed in MIL-STD-1521, *Technical Reviews and Audits for Systems, Equipment and Computer Software*.

Section 2 of the Guide discusses fault tolerant design methodology consisting of the development of C^3I program requirements, the creation of a baseline design and the systematic introduction of fault tolerance to meet R/M/T requirements.

Section 3 contains a discussion of tasks to aid the technical manager in planning, managing and tailoring R/M/T programs for fault tolerant C^3I system development. Program tailoring is the process by which individual requirements are evaluated to determine suitability for a particular system development and acquisition. The tailoring approach recommended has evolved from an extensive review of applicable military standards governing the conduct of R/M/T and system safety programs for systems and equipment.

Section 4 discusses R/M/T specification practices and the process for developing mission and safety critical fault tolerant system requirements.

Section 5 discusses the interrelationships between R/M/T in the fault tolerant design process, and includes a discussion of accessibility, fault isolation, maintenance technician training/skill levels, and design strategies for fault tolerant maintainable designs. Also discussed are the need to incorporate testability provisions, descriptions of testability design techniques, testing in the presence of faults, fault detection latency times, and partitioning for fault isolation.

Section 6 of the Guide contains a discussion of various hardware and software fault tolerant design options available to designers. The advantages, disadvantages, and R/M/T impacts of these design techniques are discussed, along with issues related to fault detection, fault avoidance, distributed processing, and levels of redundancy implementation.

Achievement of fault tolerance often requires the addition of switching devices, error detectors and other peripheral devices. The technical manager must ensure that potential reliability gains are not offset by increased failure rates of these devices.

Section 7 provides background information and describes the methodology for evaluating designs and conducting tradeoff analyses.

Sources of information used in this study included: DOD directives, NASA, DOD and military standards, military handbooks, open literature, and RADC technical reports on R/M/T for fault tolerance. Appendix A contains a list of acronyms; Appendix B a glossary of R/M/T and fault tolerance terms; and Appendix C contains a list of military documents used to develop this Guide, and a reference list.

Figure 1-1 provides a convenient quick reference to important topics discussed in this Guide.

Topic	Section	Topic	Section
Design methodology	2	Design options	
		• Hardware redundancy	
Program tailoring		– Active	6.1.1
• Reliability	3.1	– Standby	6.1.2
• Maintainability	3.2	– Voting	6.1.3
• Testability	3.3	– K of N	6.1.4
• Software	3.4	– Dynamic	6.1.5
		– Hybrid	6.1.5.1
Rqmts/spec formulation		– Adaptive voting	6.1.5.2
• Fault tolerance	4.1	– Pooled spares	6.1.5.3
• R/M/T	4.2	– Graceful degradation	6.1.5.4
• Verification	4.3		
• Warranties	4.4	• Software fault tolerance	6.2
R/M/T interrelationships		• Fault avoidance techniques	6.3
• Rel impact on M&T	5.1		
• Maintainability concept	5.2	• Distributed processing	6.4
– Maintenance trades	5.2.1		
– Scheduled maintenance	5.2.2	• Levels of implementation of	
– Deferred maintenance	5.2.3	fault tolerance	6.5
– Design criteria	5.2.5		
– Accessibility	5.2.6	Evaluation/trades	
– Maintenance personnel	5.2.7	• FMEAs	7.1
• Testability	5.3	• R/M/T models	7.2
– Testability concepts	5.3.1	– Markov models	7.2.1
– Manual test	5.3.1.1	– Model limitations	7.2.2
– Automatic test	5.3.1.2	• Readiness/availability	7.3
– Built-in-test	5.3.1.3	• Effectiveness analysis	7.4
– Testability designs	5.3.2	• Logistics analysis	7.5
– Testability impacts	5.3.2.1	• Life cycle cost analysis	7.6
– Testability considerations	5.3.2.2		
– Testability guidelines	5.3.2.3	Checklist questions	
– Test methodology	5.3.2.4	• R/M/T specifications	4.5
– Fault latency time	5.3.2.5	• Maintainability	5.2.8
– Partitioning	5.3.2.6	• Testability	5.3.3
		• Hardware redundancy	6.1.6
		• Software	6.2.4
		• Fault avoidance	6.3.6
R89-0667-001		• R/M/T evaluation	7.7

Figure 1-1. Quick Reference Index to Important R/M/T Topics.

2 - FAULT TOLERANT DESIGN METHODOLOGY

The need for fault tolerance and the selection of specific fault tolerance design techniques is highly dependent on the intended application of the system. Probability of system success, availability, and reduced downtime are key fault tolerance drivers in many C^3I applications, where a hardware or software fault could interfere with continued system operation - with potential catastrophic results. For applications with relatively short missions or high short term availability requirements, continuous fault masking (see Section 6) may be adequate, and reconfiguration (or automatic replacement of the faulty resource) may not be required. However, long duration missions or situations where high long term availability is needed, typically require failure detection and isolation, followed by the use of redundant hardware elements, reconfiguration of system elements, or the use of functionally redundant system elements. This action is required to preclude any adverse effect on system operation caused by the faulty component.

Other large C^3I systems embody processing systems where error-free computing is required or where the critical element often is the database. Here fault tolerant design techniques are required to guard against errors and protect the database. Thus, fault tolerant design techniques should be used in these systems to:

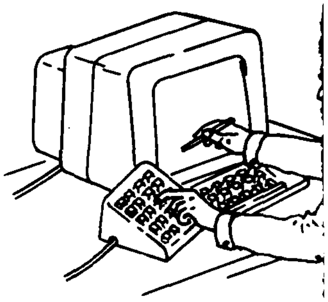
- Provide for continued, uninterrupted, operation in the presence of faults (i.e., provide high probability of success)
- Minimize damage caused by a failure (i.e., provide for fail safe operation or prevent propagation of errors)
- Minimize system downtime by permitting continued operation (possibly in a degraded mode) while the system is repaired (i.e., improve availability).

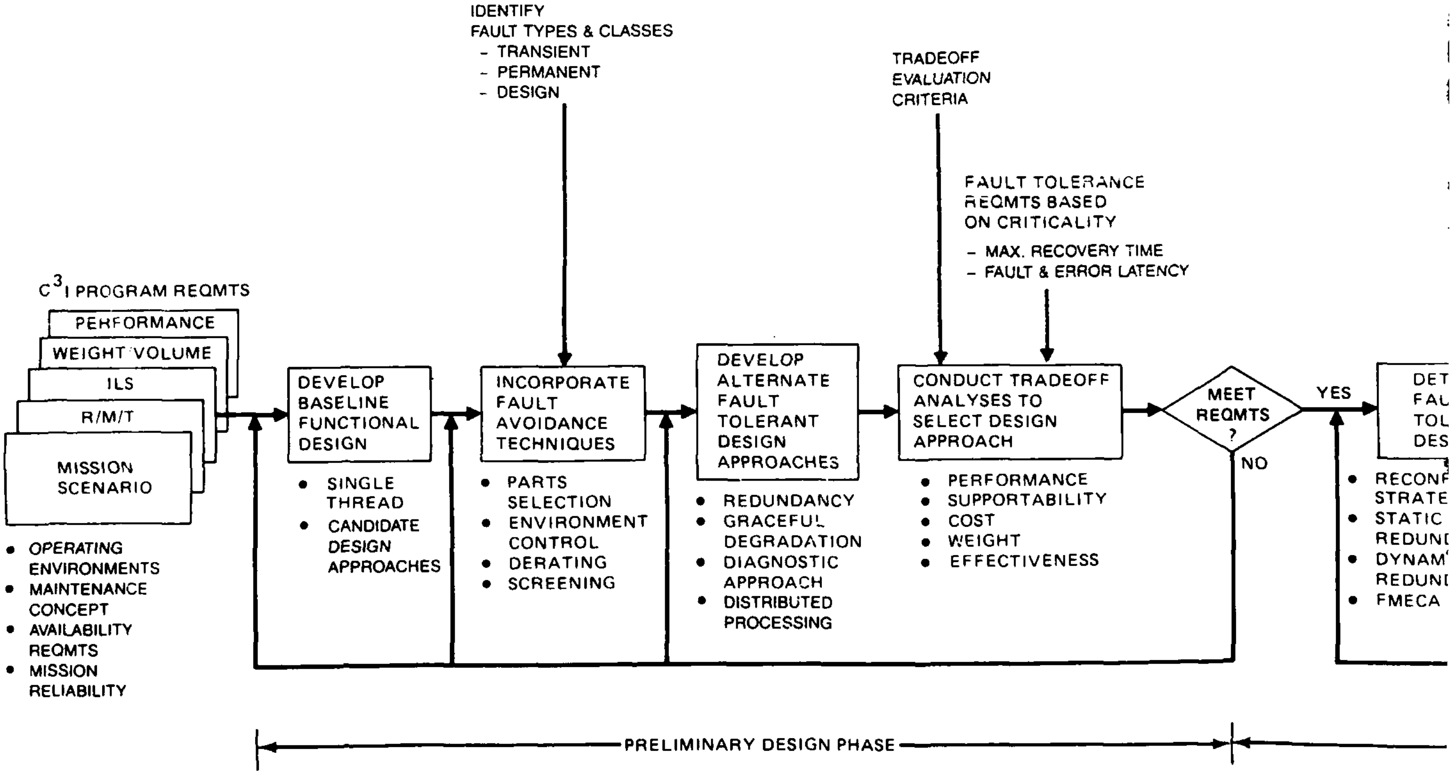
Fault tolerance must be incorporated into the design as part of the system engineering process. Figure 2-1 shows the recommended fault tolerant design methodology. The approach consists of developing C^3I program requirements, creating a baseline design, and systematically introducing fault tolerance to meet the R/M/T requirem^nts. This assures that fault tolerant design emphasis is directed at critical areas and not applied indiscriminately across the entire system. The design process is iterative and assures that all system requirements can be achieved within program cost and schedule constraints.

The first step in the fault tolerant system design process is to develop C^3I program requirements. To the extent practicable these requirements should be developed during the Concept Exploration Phase and should include system performance, weight/volume considerations, identification of the preferred maintenance concept, R/M/T and availability considerations, expected operating environments, and an established mission scenario. Although little hard data may be available during the Concept Exploration Phase, parametric and qualitative analyses can be conducted to permit these requirements to be developed early. This will assure that the production system contains the fault tolerant capability and R/M/T attributes necessary for the C^3I application. All requirements are further evaluated and refined during the Demonstration/Validation Phase.

The next step in the fault tolerant design process is to develop a baseline system architecture for the implementation technology that meets the system performance requirements. This first-cut architecture is usually non-redundant; i.e., it contains only the minimum hardware complement needed to meet the performance parameters. Furthermore, technology used in the baseline design should represent a reasonable and attainable development risk that is consistent with the program cost and schedule constraints. The use of high risk technology that is incompatible with program cost and schedule will inevitably result in serious R/M/T and system performance deficiencies.

After the baseline design has been developed, applicable fault avoidance techniques should be identified and carefully evaluated. These





R89-0687-002
R87-3537-017(T)
R88-7339-003

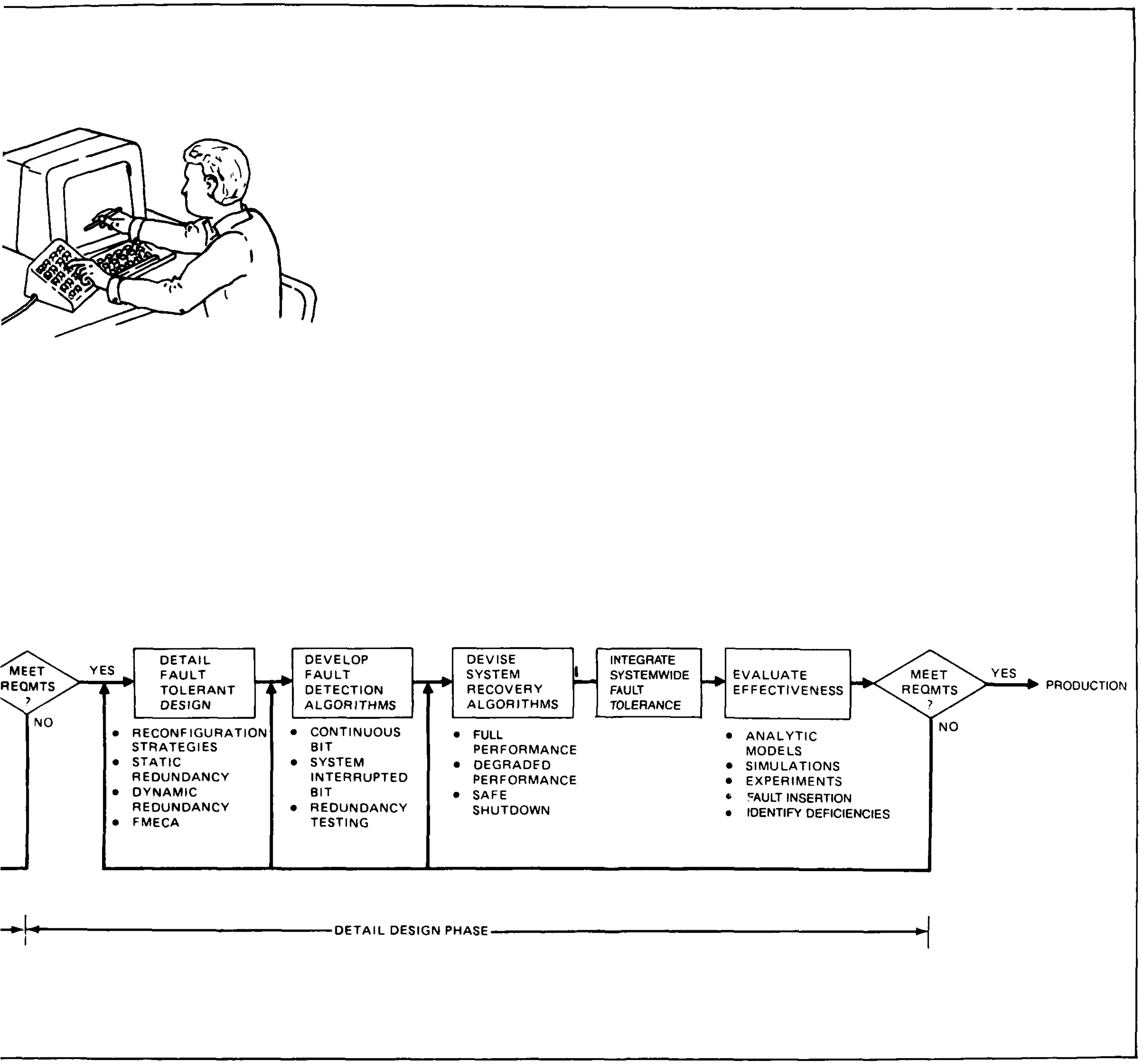


Figure 2-1. Fault Tolerance Design Methodology.

techniques normally represent the most cost-effective method of increasing system reliability. Typically, they include the following approaches:

- Reduction of environmental stresses by providing increased cooling and/or vibration isolation. For example, at operating temperatures between 10°C and 50°C, a 10% to 15% percent increase in electronic equipment reliability can be expected for each 10°C decrease in temperature
- Use of military grade piece parts instead of commercial grade
- Application of a more stringent part derating policy for new designs
- Imposition of environmental stress screening (ESS) at the piece part and equipment levels.

In general, applying fault avoidance techniques to the baseline design will maximize the system's potential field reliability.

Alternate fault tolerant design approaches that take into consideration redundancy, graceful degradation, and diagnostic schemes should then be developed. Typically, several candidate designs are initially configured and qualitatively evaluated against the major system drivers, i.e., performance, cost, weight, supportability, etc. The spectrum of alternate configurations considered should include derivatives of the non-redundant baseline in addition to innovative "new look" configurations that incorporate state-of-the-art design concepts. These alternate configurations should consider design techniques such as distributed processing and redundancy, and should include provisions for graceful degradation and fault diagnostics.

Tradeoff analyses among the various candidate design approaches are then conducted. Evaluation criteria for the tradeoff analysis should include:

- System performance
- Weight/power/volume
- Life cycle cost
- System effectiveness
- Supportability (including reliability and maintainability).

The ability of the candidate designs to meet stringent recovery time and fault/error latency requirements should be considered. Normally, the two most promising candidate approaches are selected for further configuration definition and tradeoff analysis. Alternate testability/diagnostic concepts should be evaluated as part of the design tradeoff process. System level failure mode, effects, and criticality analyses should be conducted on each alternate candidate configuration to identify single-point failures and other potential design weaknesses that impact safety and reliability.

The preferred preliminary design configuration should then be subjected to a rigorous analysis to determine whether it meets all applicable system and subsystem requirements. If the preliminary design selected does not meet R/M/T requirements, the preliminary design process must continue first by applying more stringent R/M/T design techniques in an attempt to alleviate the deficiency and, if necessary, by considering a wider range of design alternatives. If it becomes apparent that the R/M/T requirements are not achievable, the system requirements must be reevaluated and the contractor technical manager should recommend alternate requirements that satisfy the overall system objective.

In general, design trades should continue long after the preliminary design review and focus on the detail design issues. During the detail design phase the preferred preliminary design configuration is further developed and refined. Design activity should be directed at developing reconfiguration strategies, analysis of standby and active redundancy alternatives, analysis of environmental factors that affect component reliability, and incorporating design features that facilitate maintenance. Fault detection algorithms must be developed and emphasis placed on testing redundant elements in the system. Schemes for both continuous and system-interrupted built-in-test (BIT) should be refined and test frequencies established based upon an analysis of function criticality, maximum reconfiguration times, and system overhead penalties associated with BIT. These schemes are then coupled with system fault tolerance features to enable recovery and reconfiguration (if warranted) at the appropriate local and/or global level.

A vital link in the methodology is a detailed failure mode, effects, and criticality analysis. This is conducted to identify potential design weaknesses, classify each potential failure mode according to severity, and confirm fault detection/isolation features. System recovery algorithms should be devised that enable transitions to degraded modes of operation or safe shutdown when redundant resources have been exhausted due to failure.

Finally, the effectiveness and ability of the design to meet the system requirements must be reevaluated. Commonly used evaluation techniques include analytic models, simulations, experiments and demonstrations. Like the preliminary design phase, the detail design activities can be iterative processes in the event that certain R/M/T requirements have not been met.

3 - R/M/T & SOFTWARE PROGRAM TAILORING

This section contains a discussion of tasks to aid the technical manager in the tailoring of reliability, maintainability, testability, and software programs for fault tolerant systems. A more generalized discussion of R/M/T and software program tailoring, including detailed task descriptions, flow diagrams and areas requiring special emphasis for fault tolerant systems may be found in Reference 32.

In general, the R/M/T tasks and associated task application matrices contained in MIL-STD-785, MIL-STD-470, and MIL-STD-2165 adequately describe the tasks required for developing fault tolerant systems. However, the technical manager should be aware that some of these tasks require additional emphasis and tailoring for fault tolerant system developments. Task tailoring depends upon the extent of new design and development involved as well as performance level requirements. A reduced set of R/M/T tasks might be appropriate and cost effective for fault tolerant programs that use existing or commercial equipment.

As a minimum, technical managers should include the following tasks when tailoring R/M/T programs for fault tolerant applications:

- R/M/T program plans
- Allocation of specification requirements
- Design criteria
- Trade studies
- Thermal design analysis
- R/M/T predictions
- Test/verification planning
- Environmental stress screening
- In-depth design reviews
- Built-in-test analysis
- Operational assessment.

The remaining paragraphs in this section discuss (from the perspective of the technical manager) those tasks requiring special emphasis for fault tolerant systems.

3.1 Reliability Program Tailoring

The MIL-STD-785 task application matrix, which has been modified for fault tolerant system developments, is shown in Fig. 3-1. It lists applicable reliability tasks that the technical manager should consider, and recommends various references available for more detail on these tasks.

In general, the technical manager should refer to MIL-STD-785 before tailoring reliability programs to fault tolerant systems. Many of these tasks are implemented in the same way for both fault tolerant and non-fault tolerant systems but should be implemented earlier in the development process for fault tolerant systems. When developing reliability programs for fault tolerant systems the technical manager should place emphasis on the following tasks:

- *Task 101, Reliability Program Plan* - The reliability program plan should reflect system level fault tolerance requirements listed in the SOW and the system specification. **The plan should describe efforts necessary to develop specific procedures for evaluating and demonstrating how well the design meets applicable fault tolerance requirements** (including fault protection coverage and fault recovery times)
- *Task 201, Reliability Modeling* - The reliability model should identify all redundant elements (series/parallel, active, standby, pooled spares, etc.) and hardware switching elements (i.e., voters) necessary to control redundant hardware elements. **The reliability model must include shared resources or pooled spares if fault tolerance is to be enhanced by reconfiguration of these elements**
- *Task 203, Reliability Predictions* - Performing reliability predictions early in the system development process will identify equipment with inadequate stress margins. Furthermore, review

TASK	TITLE	TASK TYPE	PROGRAM PHASE				REFERENCE DOCUMENTS
			CONCEPT	VALID	FSED	PROD	
101	RELIABILITY PROGRAM PLAN	MGT	G	G	G	G	MIL-STD-785 REF 32 ¶2.1.1
102	MONITOR/CONTROL OF SUBCONTRACTORS & SUPPLIERS	MGT	S	S	G	G	MIL-STD-785
103	PROGRAM REVIEWS	MGT	S	S(2)	G(2)	G(2)	MIL-STD-1521
104	FAILURE REPORTING, ANALYSIS & CORRECTIVE ACTION SYSTEM (FRACAS)	ENGRG	NA	S	G	G	MIL-STD-2155
105	FAILURE REVIEW BOARD (FRB)	MGT	NA	S(2)	G	G	MIL-STD-785
201	RELIABILITY MODELING	ENGRG	G	G(2)	G(2)	GC(2)	MIL-STD-756 REF 32 ¶2.1.1
202	RELIABILITY ALLOCATIONS	ACCT	G	G	G	GC	MIL-HDBK-338
203	RELIABILITY PREDICTIONS	ACCT	G(2)	G(2)	G(2)	GC(2)	MIL-HDBK-217 REF 32 ¶2.1.1 MIL-STD-756
204	FAILURE MODES, EFFECTS, & CRITICALITY ANALYSIS (FMECA)	ENGRG	G(1)(2)	G(1)(2)	G(1)(2)	GC(1)(2)	MIL-STD-1629 SECTION 7.1
205	SNEAK CIRCUIT ANALYSIS (SCA)	ENGRG	NA	NA	G(1)	GC(1)	MIL-HDBK-338
206	ELECTRONIC PARTS/CIRCUITS TOLERANCE ANALYSIS	ENGRG	NA	NA	G	GC	MIL-HDBK-338
207	PARTS PROGRAM	ENGRG	S	S(2)	G(2)	G(2)	MIL-STD-965
208	RELIABILITY CRITICAL ITEMS	MGT	S(1)	G(1)	G	G	MIL-HDBK-338
209	EFFECTS OF FUNCTIONAL TESTING, STORAGE, HANDLING, PACKAGING, TRANSPORTATION & MAINTENANCE	ENGRG	NA	S(1)	G	GC	MIL-STD-781
301	ENVIRONMENTAL STRESS SCREENING (ESS)	ENGRG	NA	S	G	G	MIL-STD-810 MIL-STD-2164
302	RELIABILITY DEVELOPMENT/GROWTH TESTING	ENGRG	NA	S(2)	G(2)	NA	MIL-STD-781
303	RELIABILITY QUALIFICATION TEST (RQT) PROGRAM	ACCT	NA	S(2)	G(2)	G(2)	MIL-STD-781 REF 32 ¶2.1.1
304	PRODUCTION RELIABILITY ACCEPTANCE TEST (PRAT) PROGRAM	ACCT	NA	NA	S	G(2)	MIL-STD-781

NOTE: PROGRAM PHASE APPLICABILITY CHANGES FROM TABLE A-1 OF MIL-STD-785 ARE SHOWN WITH HIGHLIGHTED BACKGROUND

CODE DEFINITIONS

TASK TYPE

ACCT – RELIABILITY ACCOUNTING

ENGRG – RELIABILITY ENGINEERING

MGT – MANAGEMENT

PROGRAM PHASE

S – SELECTIVELY APPLICABLE

G – GENERALLY APPLICABLE

GC – GENERALLY APPLICABLE TO DESIGN CHANGES ONLY

NA – NOT APPLICABLE

(1) REQUIRES CONSIDERABLE INTERPRETATION OF INTENT TO BE COST EFFECTIVE

(2) MIL-STD-785 IS NOT THE PRIMARY IMPLEMENTATION REQUIREMENT. OTHER MIL-STDS OR STATEMENT OF WORK REQUIREMENTS MUST BE INCLUDED TO DEFINE THE REQUIREMENTS.

R89-0687-003

Figure 3-1. MIL-STD-785 Reliability Task Application Matrix for Fault Tolerant Systems.

of early predictions might identify the need for additional hardware fault tolerance

- *Task 204, Failure Mode, Effects and Criticality Analysis (FMECA)* - Where multiple layers of redundancy or reconfiguration capability in response to failures are provided, the FMECA activity should include a review of testability features to assure that adequate fault detection/fault isolation capability exists to preclude fault propagation and support system reconfiguration
- *Task 208, Reliability Critical Items* - Since the failure of elements that control redundant hardware (e.g., switches, voters, etc) can significantly impact system reliability, safety, and availability, these elements should be treated as reliability critical items
- *Task 303, Reliability Qualification Test Program* - Technical managers should consider selectively supplementing mean time between failure (MTBF) reliability qualification tests for fault tolerant system equipment by requiring verification of mean time between critical failure (MTBCF) requirements by demonstration test. This recommendation applies to mission/safety critical subsystems/ systems which contain redundant equipment with low MTBFs. The presence of high MTBCF values, or low volume production, may make it impossible to demonstrate the MTBCF with statistical confidence. In these cases, the technical manager should assure that the MTBCF requirement has been verified by rigorous analysis that includes (as appropriate) the use of a proven reliability model (see Section 7.2) and/or Monte Carlo simulation techniques (Ref 35).

3.2 Maintainability Program Tailoring

The technical manager should refer to MIL-STD-470 before tailoring maintainability programs to fault tolerant systems. The MIL-STD-470 task application matrix, modified for fault tolerant system developments, is shown in Fig. 3-2. The matrix lists applicable tasks that the technical manager should consider, and recommends various references available for more detailed information. A number of tasks identified in this figure

TASK	TITLE	TASK TYPE	PROGRAM PHASE					REFERENCE DOCUMENTS
			CONCEPT	VALID	FSD	PROD	OPER SYSTEM DEV (MODS)	
101	MAINTAINABILITY PROGRAM PLAN	MGT	G(3)	G(3)	G	G(3)(1)	C	MIL-STD-470 REF 32 ¶2.1.2
102	MONITOR/CONTROL OF SUBCONTRACTORS AND VENDORS	MGT	N/A	S	G	G	S	MIL-STD-470
103	PROGRAM REVIEWS	MGT	S	G(3)	G	G	S	MIL-STD-1521
104	DATA COLLECTION, ANALYSIS AND CORRECTIVE ACTION SYSTEM	ENG	N/A	S	G	G	C	MIL-STD-470
201	MAINTAINABILITY MODELING	ENG	S	S(4)	G	C	C	MIL-HDBK-338
202	MAINTAINABILITY ALLOCATIONS	ACC	G(3)	G(3)	G(3)	C	C	MIL-STD-470
203	MAINTAINABILITY PREDICTIONS	ACC	S(3)	S(3)	G(2)	C	C	MIL-HDBK-472
204	FAILURE MODES AND EFFECTS ANALYSIS (FMEA) MAINTAINABILITY INFORMATION	ENG	N/A	S(2)(3) (4)	G(1)(2)	C(1)(2)	C	MIL-STD-1629 SECTION 7.1
205	MAINTAINABILITY ANALYSIS	ENG	S(3)	G(3)	G(3)	C	C	MIL-HDBK-338 SECTION 5.2
206	MAINTAINABILITY DESIGN CRITERIA	ENG	S(3)	S(3)	G	C	C	MIL-STD-2084 SECTION 5.2
207	PREPARATION OF INPUTS TO DETAILED MAINTENANCE PLAN AND LOGISTICS SUPPORT ANALYSIS (LSA)	ACC	N/A	S(2)(3)	G(2)	C(2)	C	MIL-STD-1388
301	MAINTAINABILITY DEMONSTRATION (MD)	ACC	N/A	S(2)	G(2)	C(2)	S(2)	MIL-STD-471

NOTE: PROGRAM PHASE APPLICABILITY CHANGES FROM TABLE A-1 OF MIL-STD-470 ARE SHOWN WITH HIGHLIGHTED BACKGROUND.

CODE DEFINITIONS

TASK TYPE	PROGRAM PHASE
ACC - MAINTAINABILITY ACCOUNTING	S - SELECTIVELY APPLICABLE
ENG - MAINTAINABILITY ENGINEERING	G - GENERALLY APPLICABLE
MGT - MANAGEMENT	C - GENERALLY APPLICABLE TO DESIGN CHANGES ONLY
	N/A - NOT APPLICABLE

(1) REQUIRES CONSIDERABLE INTERPRETATION OF INTENT TO BE COST EFFECTIVE.

(2) MIL-STD-470 IS NOT THE PRIMARY IMPLEMENTATION DOCUMENT. OTHER MIL-STDS OR STATEMENT OF WORK REQUIREMENTS MUST BE INCLUDED TO DEFINE OR RESCIND THE REQUIREMENTS. FOR EXAMPLE MIL-STD-471 MUST BE IMPOSED TO DESCRIBE MAINTAINABILITY DEMONSTRATION DETAILS AND METHODS.

(3) APPROPRIATE FOR THOSE TASK ELEMENTS SUITABLE TO DEFINITION DURING PHASE.

(4) DEPENDS ON PHYSICAL COMPLEXITY OF THE SYSTEM UNIT BEING PROCURED, ITS PACKAGING AND ITS OVERALL MAINTENANCE POLICY.

R89-0887-004

Figure 3-2. MIL-STD-470 Maintainability Task Application Matrix for Fault Tolerant Systems.

should be implemented at an earlier phase for fault tolerant system developments, and are so indicated.

Effective and feasible concepts for maintainability, diagnostics and maintenance must be developed and applied to ensure that alternatives are examined for overall impact on performance and life cycle cost (LCC) before the system design is finalized. When developing maintainability programs for fault tolerant systems, the technical manager should place emphasis on the following tasks:

- *Task 101, Maintainability Program Plan* - Since diagnostics can provide multiple capabilities for redundancy management, fault tolerance, performance monitoring, and basic maintenance fault localization functions, the development of a maintainability program plan is an integral part of a successful fault tolerant system development
- *Task 201, Maintainability Modeling* - For fault tolerant systems designed for an on-line maintenance concept, the maintainability modeling task must consider the effect of on-line maintenance on system performance and the ability to meet overall reliability and maintainability requirements
- *Task 205, Maintainability Analysis* - The maintainability analysis task is particularly important if the fault tolerant system is to be designed for on-line maintenance. The technical manager should assure that design provisions enable on-line maintenance to be performed and that sufficient opportunity for corrective maintenance is available during scheduled system shut-downs (if applicable)
- *Task 206, Maintainability Design Criteria* - Maintainability design criteria must be established to enable on-line maintenance (if applicable) on fault tolerant systems. These criteria should consider both the maintenance and operational environments
- *Task 301, Maintainability Demonstration* - If the fault tolerant system requirement dictates that system operation continue while a redundant subsystem is being maintained, the technical manager should require that this maintainability feature be demonstrated.

3.3 Testability/Diagnostic Program Tailoring

The tasks for establishing a testability program are the same regardless of whether the system has fault tolerant provisions. These tasks are listed in Fig. 3-3 along with a list of reference documents available for more detailed information. The technical manager should place additional emphasis on the following tasks during fault tolerant system developments:

TASK	TITLE	PROGRAM PHASE				REFERENCE DOCUMENTS
		CONCEPT	D & V	FSD	PROD	
101	TESTABILITY PROGRAM PLANNING	S	G	G	NA	MIL-STD-2165 REF 32 ¶2.1.3
102	TESTABILITY REVIEWS	G	G	G	S	MIL-STD-1521 MIL-STD-1388
103	TESTABILITY DATA COLLECTION AND ANALYSIS PLANNING	NA	S	G	G	MIL-STD-2165
201	TESTABILITY REQUIREMENTS	G	G	G	NA	MIL-STD-2165 MIL-STD-1388
202	TESTABILITY PRELIMINARY DESIGN AND ANALYSIS	NA	S	G	S	MIL-STD-2165
203	TESTABILITY DETAIL DESIGN AND ANALYSIS	NA	S	G	S	MIL-STD-2165
301	TESTABILITY DEMONSTRATION	NA	S	G	S	MIL-STD-2165 MIL-STD-471

NOTE: PROGRAM PHASE APPLICABILITY CHANGE FROM TABLE I, APPENDIX A OF MIL-STD-2165 IS SHOWN WITH HIGHLIGHTED BACKGROUND

CODE DEFINITIONS

CONCEPT - CONCEPT EXPLORATION	S - SELECTIVELY APPLICABLE TO HIGH RISK ITEMS DURING D&V, OR TO DESIGN CHANGES DURING PROD.
D&V - DEMONSTRATION & VALIDATION	
FSD - FULL SCALE DEVELOPMENT	G - GENERALLY APPLICABLE
PROD - PRODUCTION & DEPLOYMENT	NA - NOT APPLICABLE

R89-0687-005

Figure 3-3. MIL-STD-2165 Testability Task Applications Matrix for Fault Tolerant Systems

- *Task 101, Testability Program Planning* - The testability program plan should identify the methodology used to establish qualitative and quantitative requirements for fault tolerant systems. The plan should either be developed as part of the system engineering program plan (required by MIL-STD-499) or prepared as a separate document

- *Task 201, Testability Requirements* - Activities performed as part of this task should establish and identify the risks and uncertainties involved in determining objectives for performance monitoring, BIT, repair verification, fault detection/isolation, test points, and off-line test objectives
- *Task 202, Testability Preliminary Design and Analysis* - The technical manager must assure that testability design techniques are closely coordinated with the fault tolerant design process. Independent testing of redundant circuitry, fault assessment, reconfiguration into degraded modes of operation, and configuration verification should make maximum use of existing hardware and functional redundancy. The addition of hardware specifically to enable or augment testing should be held to a minimum
- *Task 301, Testability Demonstration* - The scope of the testability demonstration should be expanded and integrated with a fault tolerance verification test. The purpose of these tests is to demonstrate how well the system fault tolerant design meets requirements for fault protection coverage, fault recovery time, and false alarm constraints.

3.4 Software Program Tailoring

Software is a major system development driving element. Because software is so important in attaining system performance, fault detection, fault isolation and reconfiguration, technical managers must plan, organize, and control the software project. DOD-STD-2167 contains requirements for the development of mission-critical system software. It establishes a uniform software development process that is applicable throughout the system life cycle and incorporates practices that, based on information gathered by the DOD and industry, have been demonstrated as cost-effective. Essential software development process activities that must be considered include the following:

- Project organization and planning with special emphasis on the software development plan
- Resource estimation and allocation including cost, schedule, and staff
- Required document preparation and delivery

- Project monitoring and control
- Independent review and assessment of design
- Test and certification.

A detailed discussion of software program tailoring may be found in Ref 32.

4 - SPECIFICATION OF FAULT TOLERANT SYSTEM RELIABILITY/MAINTAINABILITY/TESTABILITY REQUIREMENTS

The specification of requirements for fault tolerant systems should be developed as soon as practicable (preferably during the Concept Exploration Phase) and refined during subsequent program phases. The major activities required during the various phases of the system acquisition process may be described as follows (Ref 33):

- *Mission Need Determination* - Identify mission area needs and conduct alternatives analysis
- *Concept Exploration* - Conduct baseline system analysis and functional baseline development; perform alternative support concept analysis
- *Demonstration and Validation* - Perform system design/operational alternatives analysis; develop firm support concept
- *Full-Scale Development* - Perform detailed mission, system, and support system analysis; develop initial support plan.

Specific R/M/T requirements that take the following factors into consideration should be addressed during each program phase:

- Probability of success (MTBCF, definition of success, etc)
- System availability
- Functional, mission, and safety criticalities (fail operational/fail safe, etc)
- Acceptable degraded modes of operation
- Inherent reliability of lower level functionally redundant elements
- Diagnostic capability commensurate with reconfiguration control (maximum reconfiguration times, fault coverage)
- Testability of the major functions (level of fault detection/ isolation, false alarm constraints)
- Maintenance concept (i.e., 1, 2, or 3 level maintenance)
- Ability to demonstrate and verify compliance with R/M/T requirements (fault protection mechanisms, manual error recovery, MTBCF, on-line maintenance, etc).

The interrelationships of these factors within the fault tolerant system design process should be considered when R/M/T requirements are developed and specified. A more comprehensive discussion of R/M/T specification practices for fault tolerant systems can be found in Ref 32.

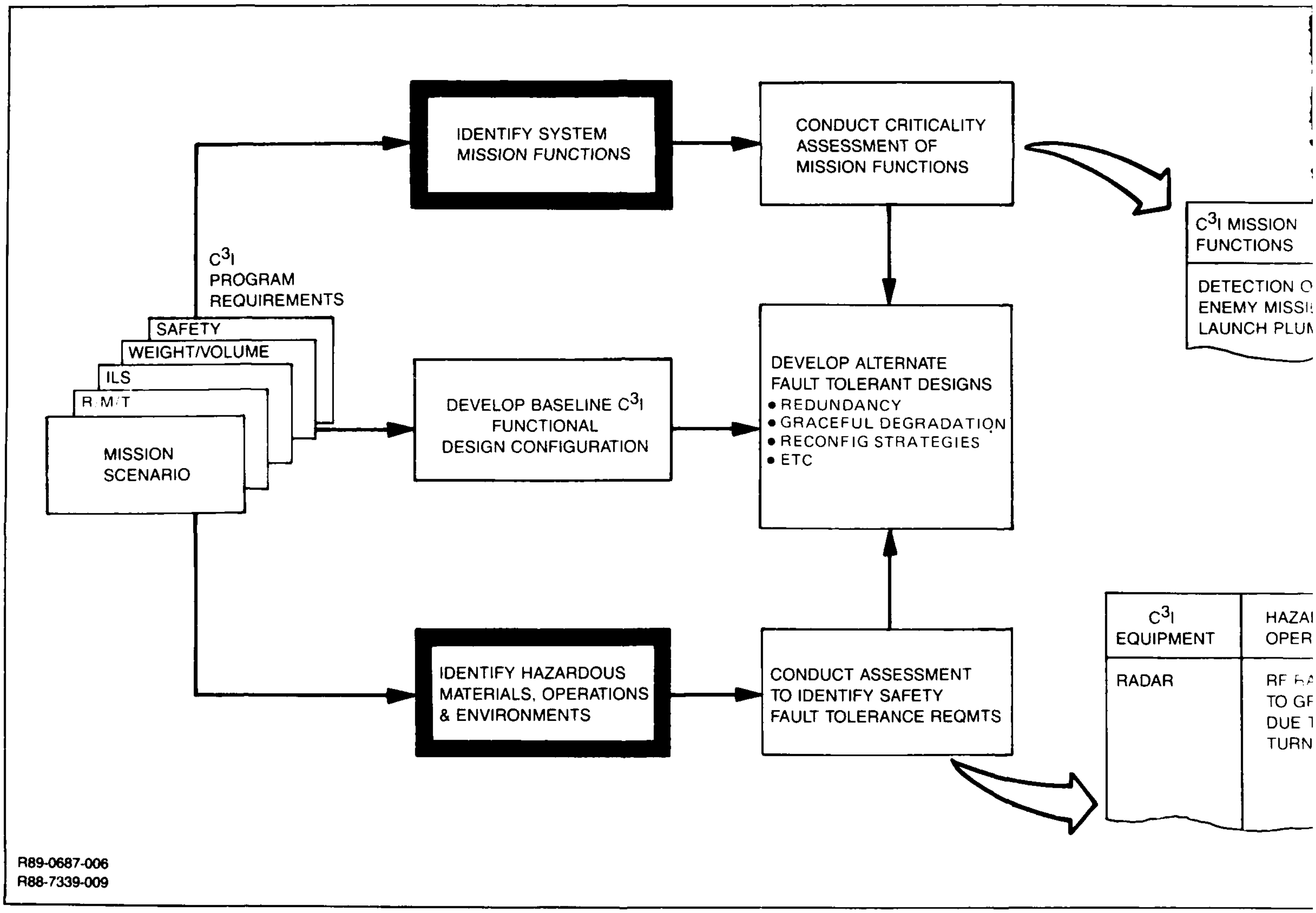
4.1 Formulation of C^3I Fault Tolerance Requirements

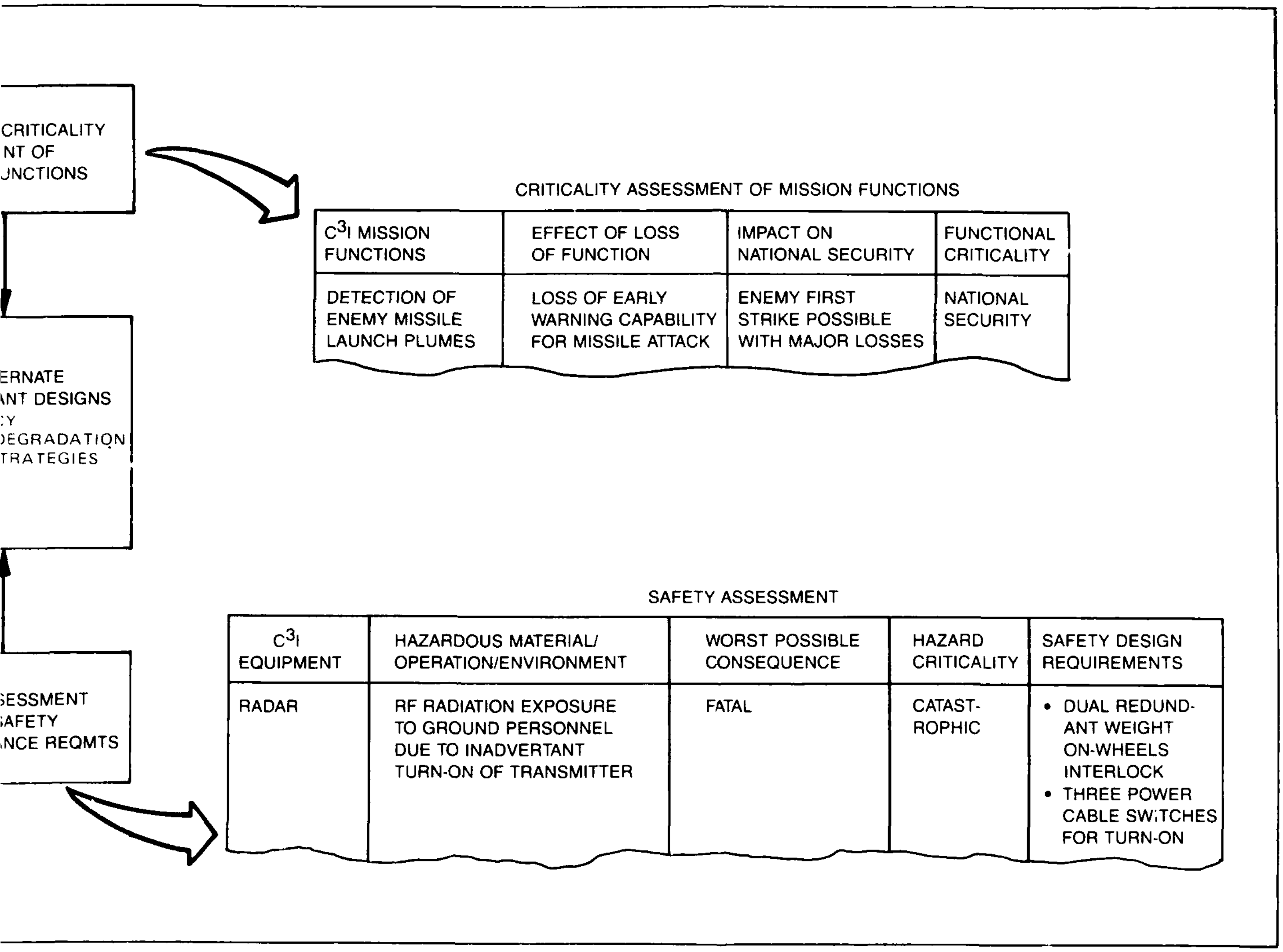
The level of fault tolerance needed in C^3I systems depends upon the operational mission, its relationship to national security and system availability and safety requirements. Fault tolerance must be judiciously implemented to avoid unnecessary program costs and logistic support requirements for spares and maintenance personnel.

Fault tolerant design implementation strategies normally are established by the contractor in compliance with the system requirements specification (from the procuring activity), and these strategies are used by designers to develop subsystem configurations. Air Force control is exercised by approval of the design concept at the preliminary design review and the detailed design at the critical design review.

Mission and safety criticality considerations generally dominate other factors when fault tolerance requirements are formulated. Figure 4-1 illustrates the process by which mission and safety-critical fault tolerance requirements are established. For mission-related requirements, the various functions of the C^3I system being considered should be identified and the consequences of a postulated loss or degradation of each function assessed. Furthermore, since the results of these assessments will form the basis for major program expenditures in manpower, equipment, development, testing, and future logistic resources, it is important that C^3I technical managers review the assessment methodology and results.

The criticality of a C^3I function is driven by its application. The criticality ranking is a relative measure of the consequences of loss of each system function on the ability of the system to perform its intended operation. This ranking is typically developed by first listing all system functions and then ranking them relative to system operational criticality and safety concerns. The principal rationale for the ranking should be


IDENTIFY SYSTEM MISSION FUNCTIONS
CONDUCT CRITICALITY ASSESSMENT OF MISSION FUNCTIONS
C3I MISSION FUNCTIONS
DETECTION O
ENEMY MISSI
LAUNCH PLUM
C3I PROGRAM REQUIREMENTS
SAFETY
WEIGHT/VOLUME
ILS
R/M/T
MISSION SCENARIO
DEVELOP BASELINE C3I FUNCTIONAL DESIGN CONFIGURATION
DEVELOP ALTERNATE FAULT TOLERANT DESIGNS
• REDUNDANCY
• GRACEFUL DEGRADATION
• RECONFIG STRATEGIES
• ETC
IDENTIFY HAZARDOUS MATERIALS, OPERATIONS & ENVIRONMENTS
CONDUCT ASSESSMENT TO IDENTIFY SAFETY FAULT TOLERANCE REQMTS
C3I EQUIPMENT
HAZA
OPER
RADAR
RF HA
TO GR
DUE T
TURN
R89-0687-006
R88-7339-009



CRITICALITY ASSESSMENT OF MISSION FUNCTIONS

C^3I MISSION FUNCTIONS	EFFECT OF LOSS OF FUNCTION	IMPACT ON NATIONAL SECURITY	FUNCTIONAL CRITICALITY
DETECTION OF ENEMY MISSILE LAUNCH PLUMES	LOSS OF EARLY WARNING CAPABILITY FOR MISSILE ATTACK	ENEMY FIRST STRIKE POSSIBLE WITH MAJOR LOSSES	NATIONAL SECURITY

SAFETY ASSESSMENT

C^3I EQUIPMENT	HAZARDOUS MATERIAL/ OPERATION/ENVIRONMENT	WORST POSSIBLE CONSEQUENCE	HAZARD CRITICALITY	SAFETY DESIGN REQUIREMENTS
RADAR	RF RADIATION EXPOSURE TO GROUND PERSONNEL DUE TO INADVERTANT TURN-ON OF TRANSMITTER	FATAL	CATAST-ROPHIC	• DUAL REDUNDANT WEIGHT ON-WHEELS INTERLOCK • THREE POWER CABLE SWITCHES FOR TURN-ON

Figure 4-1. Formulation of Typical Mission & Safety Critical Fault Tolerance Requirements.

traceable back to the system specification, and sound engineering judgement is an integral part of the process. By establishing a hierarchy of criticality among system functions (as shown in Fig. 4-2), the designer is provided with insight as to which functions warrant the incorporation of fault tolerant design provisions.

SYSTEM FUNCTION	FUNCTIONAL CRITICALITY
WEAPON GUIDANCE	1 (HIGHEST)
ATTACK CONTROL	2
SYNTHETIC APERTURE RADAR IMAGERY	3
FIXED TARGET IDENTIFICATION	3
CLUTTER MAP	3
SMALL AREA – TARGET CLASSIFICATION	3
ATTACK PLANNING	4
SECTOR SEARCH	5
WIDE AREA SURVEILLANCE	6 (LOWEST)
R89-0687-007 R87-5011-003 R88-7339-010	

Figure 4-2. C^3I Functional Criticality Ranking for Hypothetical Radar System.

A safety assessment should be conducted, as illustrated in Fig. 4-1, to establish safety design requirements. The preliminary safety assessment should be performed during the conceptual design phase with emphasis on the early identification of fault provisions for hazardous areas. Hazard criticality should be established based on worst-case conditions and the potential for personnel injury or damage to the C^3I system. Figure 4-3 contains definitions from MIL-STD-882 that should be used to base hazard criticality. The safety engineer then establishes design safety requirements, including fault tolerance provisions, based on hazard severity, qualitative/quantitative assessment of the hazard probability, and overall C^3I program system safety requirements.

Technical managers should carefully review rationale for establishing safety related fault tolerance requirements. It may be advisable to reevaluate the C^3I program objectives, design approaches, and fault tolerance requirements in light of the safety assessment results.

DESCRIPTION	CATEGORY	MISHAP DEFINITION
CATASTROPHIC	I	DEATH OR SYSTEM LOSS
CRITICAL	II	SEVERE INJURY, SEVERE OCCUPATIONAL ILLNESS, OR MAJOR SYSTEM DAMAGE
MARGINAL	III	MINOR INJURY, MINOR OCCUPATIONAL ILLNESS, OR MINOR SYSTEM DAMAGE
NEGLIGIBLE R89-0687-008 R87-3637-011(T)	IV	LESS THAN MINOR INJURY, OCCUPATIONAL ILLNESS, OR SYSTEM DAMAGE

Figure 4-3. MIL-STD-882 Criticality Categories.

4.2 Qualitative and Quantitative R/M/T Requirements

There are two approaches to establishing qualitative and quantitative fault tolerance requirements:

- *Classical top down* - Establish mission requirements; then derive fault tolerance requirements as a function of mission, restoration, and testability design characteristics
- *Bottom up* - Define the lowest level functional element and then establish fault tolerance requirements in relationship to function criticality.

The top down approach is preferred since it permits user/operator/mission needs (which are generally available early in the system development cycle) to be translated into system/subsystem and lower tier hardware requirements. The derived requirements must be checked for realism in that they should be consistent with available technology and system constraints (i.e., weight, cost, power, volume, etc).

Subsystem and lower-level requirements must satisfy the overall allocations of system level fault tolerance requirements that were derived from the mission requirement. **Quantitative top-level fault tolerance requirements should be derived from parametric sensitivity analyses and tradeoffs to optimiza system readiness.** The process of refining and evaluating these top-level fault tolerance requirements during the design process is described in Section 7 of this Guide.

Managers should consider the following general guidelines when deriving, implementing, or responding to R/M/T system specification requirements:

a. Is the requirement overspecified? (Overspecifying will lead to higher development, test, and production costs)

b. Is the wording of the requirement subject to misinterpretation?

c. Is the requirement necessary, or is it included merely because of previous usage?

d. Can compliance with the requirement be verified? How? Through test, simulation, analysis?

e. Is the statement of the requirement consistent across system, subsystem, and other lower level groupings?

f. Have adequate design margins (tolerance) been allowed?

g. Has tailoring been considered for all referenced standards?

h. Are the R/M/T attributes of off-the-shelf equipment to be used consistent with overall C^3I system requirements?

4.2.1 Specification of Reliability Requirements

During the Concept Exploration and Demonstration/Validation Phases, technical managers must determine the permissible level of system performance degradation that can be tolerated without compromising mission success. Based upon these findings, satisfactory system performance can be defined and included in the reliability requirements section of the C^3I system specification. The technical manager must make certain that the system specification contains a clear, unambiguous statement as to which system operating modes are required and what levels must be attained for satisfactory performance.

When planning, responding to, or aiding in the preparation of reliability requirement inputs to fault tolerant C^3I system specifications, managers should consider the following:

- Critical mission definition
- Quantitative mission reliability
- Quantitative maintenance frequency reliability
- Description of storage, transportation, operation, and maintenance environments
- Time measure or mission profile
- Definition of satisfactory and acceptable degraded system performance

- **Tolerable failure policy (fail safe, fail-op/fail safe, etc)**
- **Failure independence.**

MIL-HDBK-338 provides guidelines and examples useful in preparing reliability specification inputs.

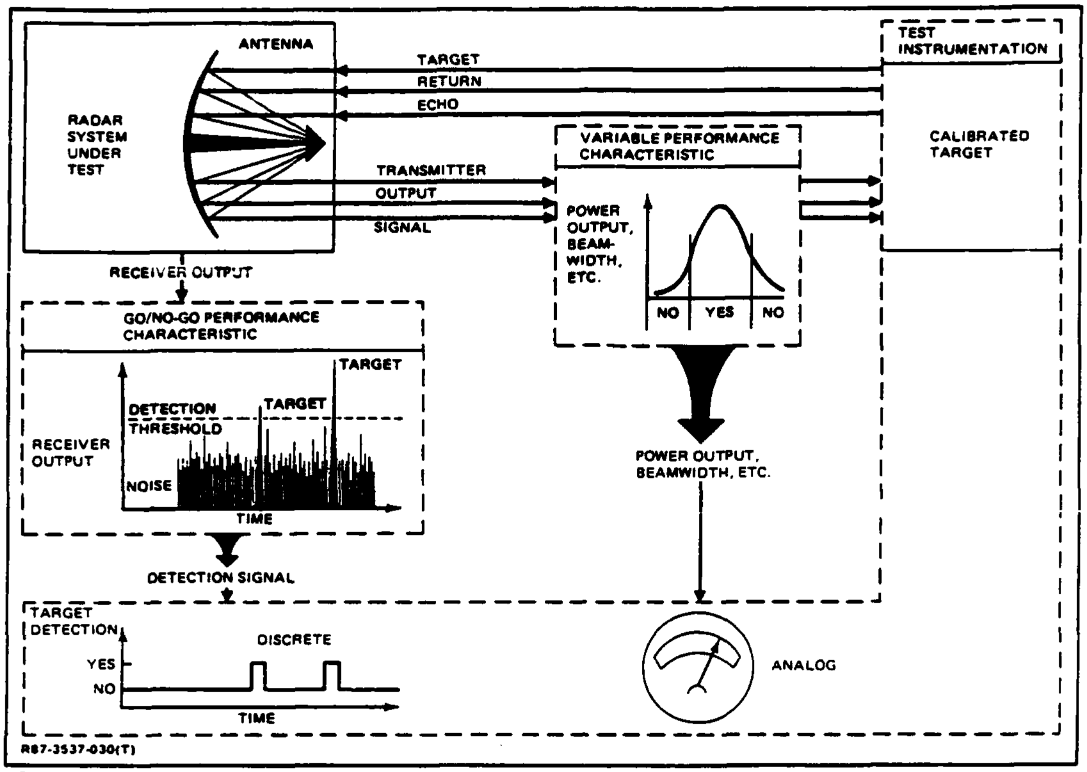
The specification of operational/field R&M requirements as being separate and distinct from contractual requirements is not unique to fault tolerant systems. However, managers are advised to pay particular attention to this distinction in view of the emphasis placed on meeting numerical R&M requirements. **In general, when specifying maintenance frequency reliability, managers should assure the following:**

- Operational/field terms must be distinguished from contractual **terms (Ref DOD Directive 5000.40 and Ref 21)**
- Numerical traceability **between** operational/field terms and contractual terms
- Consistency must be established and maintained between operational/field and contractual requirements. .

In developing a satisfactory and acceptable degraded performance level, technical managers should emphasize the following:

- Removal of any ambiguity from the interpretation of quantitative reliability requirements
- Inclusion of a clear, unequivocal definition of "failure" for the equipment/system relative to its important performance parameters.

Figure 4-4 illustrates two types of performance characteristics and corresponding success/failure (yes/no) decision boundaries that might be applied to a track radar or to an active seeker missile guidance system. In both cases, the success/failure boundary must be determined for each essential system performance characteristic measured in the demonstration test. This will minimize the chance for subjective interpretation of failure definition, and post-test rationalization (other than legitimate diagnosis) of observed failures.



R88-0887-008
R88-7339-012

Figure 4-4. Verification of System Performance Characteristics.

Failure independence requirements may stipulate fault containment or fault propagation restrictions to limit both the immediate effects of faults and possible secondary failure effects. **When specifying a tolerable failure policy or failure independence requirements, the equipment level to which the requirement applies must be specified.**

C^{3}I systems may contain many operating modes and functions, some of which are used in peacetime, and some in wartime. In such cases, it is recommended that critical mission capability and use environments that are tied to an essential mission performance level be defined for both peacetime and wartime scenarios. This definition could then be related to quantitative reliability and availability requirements and their respective demonstrations/verifications.

Figure 4-5 contains a number of samples of the language used in reliability specifications of fault tolerant systems.

EXAMPLE 1: C^3I DATA FUSION SYSTEM

RELIABILITY

FAIL SAFE DESIGN – The XYZ system shall not have any single point failures in the critical path (i.e., the design shall compensate for a failure by redundancy or an alternate operating procedure to insure the continued flow of message traffic). A critical path is defined as any path within the XYZ system which is necessary for correct data flow.

This is an example of a tolerable failure policy applicable to the system level. Also included is a reference to a critical path which defines the system's essential functional requirements.

MISSION RELIABILITY – The mission time-between-critical-failure (MTBCF) shall be no less than *xxx* hours when operated under the environmental conditions specified herein. The design of the XYZ system shall result in a *predicted* MTBCF equal to or exceeding twice *(as a rule-of-thumb)* the *specified* MTBCF. A critical failure is defined as any failure in which a critical mission capability is not restored in less than *yyy* milliseconds.

This is an example of specifying MTBCF rather than probability of mission success (R_M). In addition, the mission criticality is such that a maximum time is specified to restore (via redundancy or alternate operating procedure) the critical mission capability.

MAINTENANCE FREQUENCY RELIABILITY – The mean time between (corrective) maintenance (MTBM) actions, as defined in AFR 80-18 of the XYZ system, shall be no less the *zzz* hours. The design of the XYZ system shall result in a *predicted* MTBM equal to or exceeding twice the *specified* MTBM.

This is an example of specifying an operational or field reliability parameter as measured by the AF 66-1 Maintenance Management System. Verification or demonstration of this requirement normally would be accomplished using field data during initial deployment. Another approach could be to specify a minimum acceptable system MTBF verified by a MIL-STD-781 demonstration test.

INDEPENDENCE OF FAILURE – The XYZ system shall be designed such that a unit level failure can not induce any other failure.

CRITICAL MISSION CAPABILITY – Critical mission capability is that level of performance which shall allow the XYZ system to perform its mission of supporting the required communications and information flow without degradation. The following XYZ functions shall be operating in order for the system to meet critical mission capability...

This is an example of a complex system wherein it is necessary to tie the quantitive mission reliability requirement to an essential mission performance level.

EXAMPLE 2: FAULT TOLERANT AIRBORNE AVIONICS SYSTEM

RELIABILITY – The XYZ system shall have a predicted reliability (as specified below) based on analysis in accordance with MIL-HDBK-217. This includes all components of redundant circuits employed to achieve fault tolerance. The predicted reliability under the temperature and altitude conditions specified herein for continuous operation, shall be not less than:

a. Mean Time Between Failures (MTBF) = *xxx* hours (includes failures in redundant circuits)
b. Mission Time Between Critical Failures (MTBCF) = *yyy* hours (System Fail-Operational capability maintained) (see Level of Fault Tolerance below)

NOTE: When a malfunction is detected, it is assumed that maintenance to restore full fault tolerance capability occurs after each mission or the first available time.

MAINTENANCE FREQUENCY RELIABILITY – The mean time between corrective maintenance action of the XYZ system shall be no less than *zzz* flight hours.

R89-0687-010(1/2)
R89-7339-013(1/2)

Figure 4-5. Examples of Reliability Specification of Fault Tolerant Systems. (Sheet 1 of 2).

INDEPENDENCE OF FAILURE – The XYZ system shall be designed such that at the *ABC* (e.g., subsystem, unit, assembly, LRU, SRU) level no failure shall induce any other failure.

FAULT TOLERANCE REQUIREMENTS – The XYZ system shall provide a fault tolerant performance capability in compliance with the contractual statement of work, and in accordance with the following criteria.

LEVEL OF FAULT TOLERANCE – This capability shall specify the system (or subsystem, by function) response to any randomly occurring single fault, or sequence of unrelated faults. Failure is defined as any situation, detected by any means, in which the XYZ system does not meet specification requirements during operation. The applicable levels for a single fault are:

a. Fail-Safe: XYZ system output data frozen or disabled – failure annunciated – before any variable error exceeds 2x specified accuracy.
b. Fail-Operational: XYZ system output data continues uninterrupted – status change annunciated – transsient disturbances do not exceed 2x specified accuracies.

The response of the Fail-Operational system to each subsequent fault in a sequence shall result in no less than the following system state (provided by redundancy or a degraded operating mode):

Fail-Operational/Fail-Safe – (System state complies with Fail-Safe criteria)

FAULT PROTECTION COVERAGE – At the "FAIL-SAFE" level, the effectiveness of the XYZ system shall not be less than *(VV)* percent for a two-hour sortie.

R89-0687-010(2/2)
R88-7339-013(2/2)

Figure 4-5. Examples of Reliability Specification of Fault Tolerant Systems. (Sheet 2 of 2)

4.2.2 Specification of Fault Protection Coverage Requirements

Fault protection coverage can be stated in both quantitative and qualitative terms. In a numerical sense, fault protection coverage is the conditional probability that the system successfully recovers when a specific type of failure has occurred. A more limited, quantitative definition of fault protection coverage relates to the probability of detecting any fault. The value of fault protection coverage is often determined by using the average of the coverages for all possible classes of failures, weighted by the probability of occurrence of each fault class. Figure 4-6

EXAMPLE 1: TOP LEVEL SYSTEM SPECIFICATION

FAULT PROTECTION COVERAGE – All fault classes for the XYZ System shall be covered except for the following:

1. Generic faults which affect all processor channels in an identical manner
2. Multiple faults, i.e., faults which affect multiple processor channels simultaneously
3. Faults which occur during reconfiguration

EXAMPLE 2: PRIME ITEM DEVELOPMENT/EQUIPMENT SPECIFICATION

FAULT PROTECTION COVERAGE – The fault protection coverage of the XYZ Subsystem shall not be less than xxx percent. Fault protection coverage is the combination of the independent probabilities of fault detection (FD), fault isolation (FI), and fault recovery (FR) for all possible faults of the system.

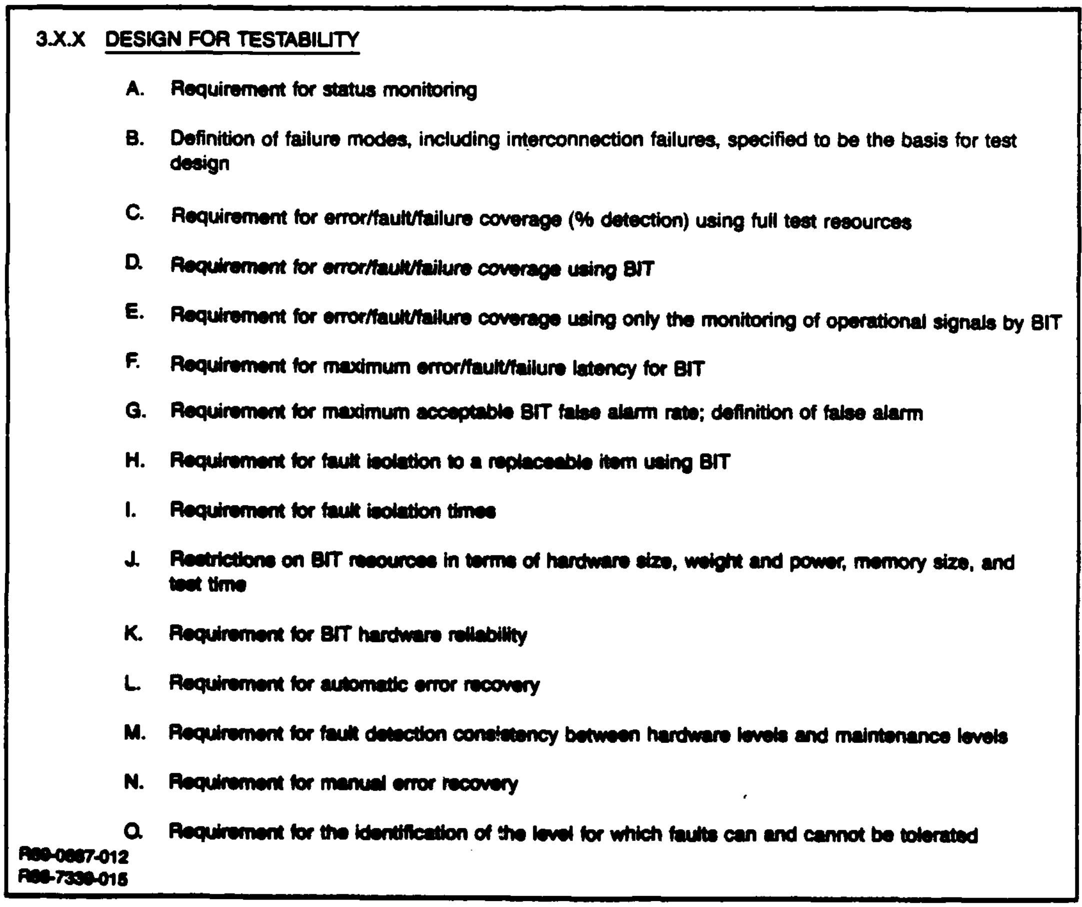
R89-0687-011
R88-7339-014

Figure 4-6. Examples of Fault Protection Coverage Specification Language.

contains examples of how fault protection coverage requirements may be specified.

4.2.3 Specification of Maintainability & Testability Requirements

Guidance for the preparation of maintainability and testability requirements are available in MIL-STD-470 and MIL-STD-2165. Figure 4-7 lists model requirements (a through n) for system testability which have been extracted from MIL-STD-2165. Requirement (n) has been added since a manual error requirement may be indicated. Automatic error recovery methods (o) such as reconfiguration, error correction codes, checkpoint rollback, redundant message sending, and/or retry may be incorporated in fault tolerant designs.



3.X.X DESIGN FOR TESTABILITY

A. Requirement for status monitoring

B. Definition of failure modes, including interconnection failures, specified to be the basis for test design

C. Requirement for error/fault/failure coverage (% detection) using full test resources

D. Requirement for error/fault/failure coverage using BIT

E. Requirement for error/fault/failure coverage using only the monitoring of operational signals by BIT

F. Requirement for maximum error/fault/failure latency for BIT

G. Requirement for maximum acceptable BIT false alarm rate; definition of false alarm

H. Requirement for fault isolation to a replaceable item using BIT

I. Requirement for fault isolation times

J. Restrictions on BIT resources in terms of hardware size, weight and power, memory size, and test time

K. Requirement for BIT hardware reliability

L. Requirement for automatic error recovery

M. Requirement for fault detection consistency between hardware levels and maintenance levels

N. Requirement for manual error recovery

O. Requirement for the identification of the level for which faults can and cannot be tolerated

R89-0687-012
R88-7339-015

Figure 4-7. Model Requirements for Testability in a System Specification.

Figure 4-8 presents a typical format covering many of the numerical requirements of Fig. 4-7 as well as many other testability and maintainability parameters of interest. This Notational Diagnostic Performance Specification (Ref 18) should be a deliverable item after both the Demonstration/Validation Phase and the FSD Phase. By accurately quantifying all the listed parameters of this specification, a meaningful assessment can be made of a fault tolerant C^3I system's testability and maintainability attributes.

LEVEL OF MAINTENANCE[1]	DIAGNOSTIC CAPABILITY[1]	FAULT DETECTION COVERAGE[2,4]	FAULT ISOLATION COVERAGE[2,4]	MEAN TIME TO DIAGNOSE	FALSE ALARM RATE[3]	FALSE REMOVAL RATE[3]	OTHER REQUIREMENTS[1]
ORGANIZATIONAL	STATUS MONITOR	______ %	______ %	______	______	______	____ % OF FAULT COVERAGE BY STATUS MONITOR FOR MISSION-CRITICAL FUNCTIONS
	BIT	______ %	______ %	______	______	______	
	MANUAL TEST	______ %	______ %	______	______	______	
	MAINT. AIDS/ MANUAL TROUBLE-SHOOTING	______ %	______ %	______	______	______	BIT MEMORY ALLOCATION NOT TO EXCEED X WORDS
	TOTAL:	100 %	100 %				TECHNICAL INFORMATION ACCESS TIME
INTERMEDIATE	EXTERNAL ATE/ EXPERT SYSTEM	______ %	______ %	______	______	______	ATE LIMITED TO X LB. Y CUBIC FT.
	MANUAL TEST	______ %	______ %	______	______	______	
	TOTAL:	100 %	100 %				
DEPOT	EXTERNAL ATE	______ %	______ %	______	______	______	
	MANUAL TEST	______ %	______ %	______	______	______	
	TOTAL:	100 %	100 %				

1. LISTED BY WAY OF EXAMPLE

2. UNAMBIGUOUS PERCENTAGE OF FAULT DETECTION COVERAGE (RATIO OF FAILURES DETECTED TO FAILURE POPULATION) FOR EACH CAPABILITY SHOWN. TOTAL AT EACH LEVEL OF MAINTENANCE SHOULD ADD TO 100% OF THE IDENTIFIED REPLACEABLE ITEMS FOR THAT LEVEL.

3. RELATE RATES TO OPERATIONAL USAGE (E.G., 1 FALSE ALARM PER MONITORING HOUR).

4. FOR EACH DIAGNOSTIC CAPABILITY LISTED, INDICATE WHETHER "P" - PRIMARY MODE OR "S" SECONDARY OR AUGMENTING MODE.

R89-0687-013
R87-5011-002
R89-7339-016

Figure 4-8. Notational Diagnostic Performance Specification.

4.3 Verification of Compliance with R/M/T Requirements

All contractual R/M/T requirements must have a contractually specified method of verifying compliance. The specification should delineate the analysis methods and demonstration tests that must be performed to verify that the specified requirement has been met. For demonstration tests, the specification should define the following:

a. How will the equipment/system be tested?

   Test conditions, environmental conditions, test measures, length of test, equipment operating conditions, accept/reject criteria, test reporting requirements, etc

b. Who will perform the tests?

   Contractor, government, or independent organization

c. When will the tests be performed?

   Development, production, or field operation phases

d. Where will the tests be performed?

   Contractor's plant, government organization, or field.

Technical managers should include a requirement in the C^3I system SOW for test and analytical methods to be identified and described in the System Test Plan, Qualification Test Plans, Engineering Development Test Plans, Testability Demonstration Plan, and Reliability Development/Growth Test Plans, as applicable. Extensive simulation and testing should be accomplished on representative high-risk hardware elements early in the development cycle and contractors should be required to document the planned approach for evaluating and demonstrating how well a design meets its specified fault tolerance goals and requirements.

4.4 Warranties

Recognizing the critical importance of warranties, Congress passed legislation in 1983 and 1984 that required the Department of Defense (DOD) to obtain warranties on major weapon systems. An update (Section 1234 of the 1985 DOD Authorization Act) provides for flexibility in structuring warranties but specifically requires them on weapon systems that have a unit cost of more than $100,000 or an expected total procurement cost of more than $10 million. It should be noted that almost all fault tolerant C^3I systems meet the above criteria. These laws were passed because of concerns that weapon systems often fail to meet their military missions, are operationally unreliable, have defective and shoddy workmanship, and can endanger the lives of the using communities.

These laws are intended to make contractors more accountable and encourage them to build better quality and reliability into their systems.

4.4.1 Reliability Improvement Warranty Planning Considerations

Due to the vital nature of C^3I systems to national defense, the importance of warranty planning for both industry and the government cannot be overemphasized. Planning should start in the weapon system concept stage. The intent of military Reliability Improvement Warranty (RIW) programs is to improve operational availability and, thus, RIWs are an important adjunct to other approaches such as modification of hardware, improvements or innovations in installation and operation, changes in maintenance procedures, or revision in logistics support policies.

In general, warranties impose important responsibilities on military organizations who must plan for, implement, and administer the RIW; likewise, the equipment manufacturer must perform a cost and risk analysis, support, monitor and accept responsibility from the inception of full-scale production, and repair all returned units for a "one time" fixed price. An RIW contract between the contractor and the procuring agency stipulates that the contractor assumes the cost of repair and/or replacement of failed equipment. It often requires that the contractor prepare and implement design changes if the equipment MTBF falls below the specified value. This responsibility often continues for a fixed period beyond the delivery of the last production unit.

The planning, wording, and administration of RIWs is similar for both fault tolerant and non-fault tolerant systems. However, the inclusion of RIWs in requests for proposals and production procurement contracts will be a major contribution to the success of complex fault tolerant military hardware programs. When required prior to seller contract award, these warranties provide a realistic basis for evaluating the seller's equipment reliability, since the seller's general response to, and particularly the pricing of, the warranty will be a direct measure of the seller's confidence in the ability of the equipment to meet the stringent R&M requirements imposed on fault tolerant systems. RIWs provide the contractor incentives and opportunities to investigate relevant anomalies and implement or recommend cost-effective changes that will assure achievement of field performance goals.

4.4.2 Contractor Incentives and Warranties

Competitively bid RIWs are worthwhile from both the procuring activity and contractor viewpoints. Military experience has proven RIWs to be very cost-effective for the procuring activities, and warranty programs motivate contractors to provide systems and equipment having the highest practical readiness capabilities without necessarily controlling the methods by which contractors design, test, or produce the system or equipment. These incentive and warranty programs focus on the essential tasks and responsibilities of the contractor, and the salient concerns of the government (e.g., operational readiness and ownership costs).

The major factors which drive warranty dollars are the number of failures per coverage period, the time required to isolate the failure, effect the repair, confirm operational restoration, labor rates, replacement parts and logistic handling costs. Technical managers desiring more detailed information on this subject should see Section II of the Air Force Electronic System Division's Readiness Improvement through Systems Engineering (RISE) Handbook. It contains details on various types of RIW programs, the associated commitments, responsibilities of parties, definition of terms, contractual clauses, and implementation data.

4.5 R/M/T Specification Checklist Questions

The following questions are intended to highlight key topics that should be considered by the technical manager for inclusion in the R/M/T requirements sections of the C^3I system specification and prime item development/equipment specifications:

a. How do the system fault tolerance requirements impact the overall reliability, maintainability, and availability requirements?

b. Have the definitions of satisfactory system performance and system failure been specified?

c. Have the maximum off-line or reconfiguration time(s) been specified or included in the definition of satisfactory performance?

d. What is the tolerable failure policy? (single point, fail safe, etc)

e. What is the required level of fault protection coverage for the system? Has fault coverage been clearly defined?

f. Have the false alarm constraints been specified?

g. Will the fault tolerance policies and methodologies be among the vital functions of the program to be evaluated and verified?

h. How will the fault protection mechanisms be demonstrated or validated?

i. Under what conditions (climatic, space, land, etc) must the system be operated and maintained? What are the maintenance strategies and concepts?

j. What functions in the system involve the most risk to mission success if they were to fail?

k. Has a requirement for manual error recovery been properly specified?

l. Has consideration been given to including an RIW requirement in specifications covering the production phase?

m. Are the fault tolerance requirements consistent with expected operational use?

- Is the normal system operation active or standby?
- What is the intended utilization cycle of the system (8 hours/day, 24 hours/day, continuous, on-demand)?
- What critical system functions warrant continuous monitoring?
- What system functions are normally active?
- What system functions are normally passive or operating in a standby mode?

n. Are the fault tolerance requirements appropriate for the operating environment?

o. What are the time constraints for BIT performance? Have they been translated into hardware requirements?

p. Have probabilistic and quantitative readiness goals been defined?

q. Have system utilization, on-station demand, and turnaround requirements been quantified?

r. Is maintenance done to the system with or without shutting it down?

s. Has a RIW program been included in the requirements specification?

5 - R/M/T IMPACT ON FAULT TOLERANT DESIGN

The life of a system is generally perceived by its users as measured in two states with respect to the specified system performance requirements:

- Proper service - where system performance is delivered as specified
- Improper service - where system performance is below specified values. (Ref 15).

Because national security considerations for C^3I systems typically require high reliability and readiness, fault tolerant systems configured to meet these needs must incorporate an optimum mix of R/M/T design features while meeting LCC constraints. Since reliability and maintainability often have competing interests (e.g., redundant equipment adds to the maintenance burden) it is important that the technical manager understands the interrelationships between R/M/T.

The success of most fault tolerant systems depends largely on the design's inherent diagnostic capability and testability; specifically, its ability to detect, identify, and report malfunctions so that suitable corrective action can be taken. The selection of a redundant design technique must include an assessment of associated diagnostic/testability alternatives and their overall impact on how well the design goal performance requirements are achieved. To assure a successful fault tolerant design, the technical manager must make certain that the designer has dealt with constraints such as cost, size and weight limitations, available power, and interface complexity restrictions.

System reliability can be improved by using redundancy techniques, but caution must be exercised in this approach. Fault detection and isolation are often the limiting factors when designing redundancy into the system. For example, a subsystem may consist of a number of redundantly configured items and the reconfiguration strategy may require isolating a failed item before an operationally redundant item can be switched into its place. Depending upon functional criticality, redundant units can be switched-in either at the first indication of a failure or after a failure indication has been sustained. In either case, once the spare unit has been switched in, the operating system can either command more exhaustive BIT testing of the faulty module and log the unit as failed if confirmed by the BIT, or return the unit to standby or active status if the failure is not confirmed. When considering additional redundancies, the technical manager should use caution since the diagnostics/testability of an item is seldom 100% perfect. **When the non-perfect probabilities of correct failure detection and isolation, together with switching time and task initiation time penalties are taken into account, it is entirely possible that the subsystem probability of mission success may not increase by adding redundant items.** The non-perfect probability of correct failure detection and isolation can be taken into account in the analysis by including an additional element (or any number of elements) in the reliability model. These elements should reflect the reliability of switching devices (see Subsection 6.1) and can also be assigned a probability of success for the fault detection and isolation (FD/FI) function. As such, the FD/FI function can be treated in the same way as a hardware element in the reliability assessment. Alternatively, Markov analysis (see Subsection 7.2.1) can be used to evaluate imperfect FD/FI by using transition rates representative of the system's fault handling characteristics. **The technical manager should carefully trade the benefits of the additional redundancy complexities and the increased maintenance burden specifically caused by diagnostic uncertainties.**

The following sections contain a discussion of the impact of R/M/T on fault tolerance, and the interactions and often conflicting interests of R/M/T that C^3I technical managers should be familiar with.

5.1 Fault Tolerant Reliability Impact on Maintainability & Testability

Reliability is a design characteristic that must be preserved during the system's operational life. **To maintain the high levels of reliability within fault tolerant systems, it is important that the system level fault tolerant features be restored to service quickly.** Hence, the same C^3I system level requirements that result in the incorporation of fault tolerance also dictate the need for maintainability and testability features.

Emphasis on maintainability and testability is increasingly important as system complexity increases. This in turn demands that diagnostics for system monitoring, control, checkout, and maintenance be integrated within the basic design. System-level R/M/T requirements (including mission reliability, type of maintenance concept, maintenance downtime restrictions, etc) upon analysis and apportionment, typically result in system and subsystem level fault tolerance requirements that include the following:

- Fail operational/fail safe levels
- Corrective maintenance strategy (on-line, off-line)
- Maximum downtime for reconfiguration or maintenance
- MTBF and MTTR apportionments.

These requirements are then translated into a series of candidate design approaches that must be evaluated and traded off to obtain the optimum approach for any given application. Each design approach (e.g., single thread, system level redundant, reconfigurable, etc) will have implications on reliability and maintainability. When reliability is measured in terms of MTBF and maintainability in terms of MRT, availability emerges as a popular metric for trade studies. Since there are practical limitations on how high a mean time between failure (MTBF) can be achieved or how low the mean time to repair (MTTR) can be made, it is often necessary to assess the interrelationships of reliability, maintainability and testability, and the limits imposed by state-of-the-art. For example, MTTR values close to zero would require unrealistic maintainability design features, such as perfect FD/FI and extremely rapid remove-and-replace times.

Since availability (see Subsection 7.3) relates to reliability, maintainability, and testability, it represents a convenient way to discuss their interrelationships. Figure 5-1 shows the relationship between MTBF, mean repair time (MRT), and availability (A_i). Because MRT is directly related to MTTR, mean logistics delay time (MLDT), and maintenance downtime (MDT), these quantities can be explored to determine the

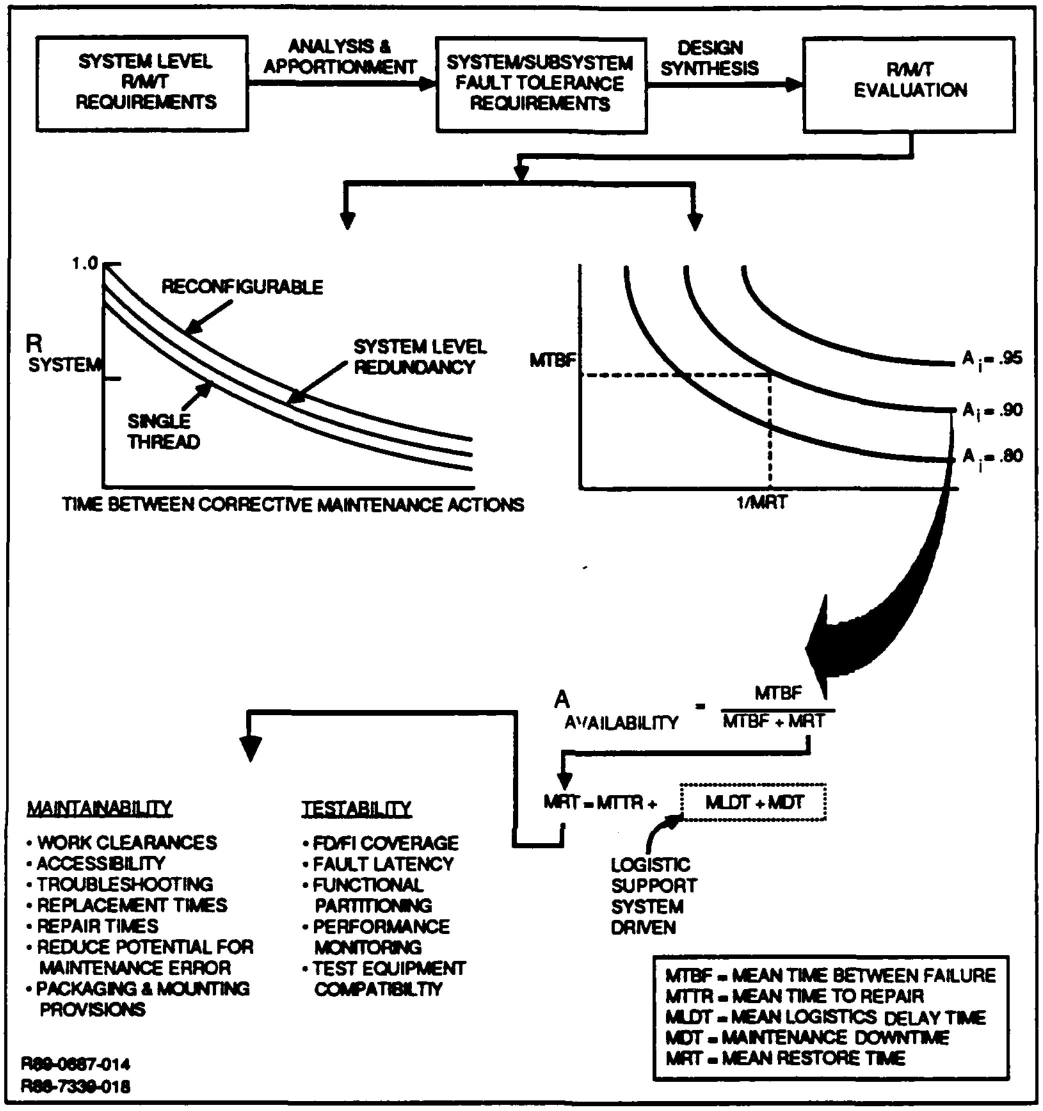

SYSTEM LEVEL R/M/T REQUIREMENTS
ANALYSIS & APPORTIONMENT
SYSTEM/SUBSYSTEM FAULT TOLERANCE REQUIREMENTS
DESIGN SYNTHESIS
R/M/T EVALUATION
1.0
R SYSTEM
RECONFIGURABLE
SYSTEM LEVEL REDUNDANCY
SINGLE THREAD
TIME BETWEEN CORRECTIVE MAINTENANCE ACTIONS
MTBF
A_i = .95
A_i = .90
A_i = .80
1/MRT
A AVAILABILITY = MTBF / MTBF + MRT
MRT = MTTR + MLDT + MDT
LOGISTIC SUPPORT SYSTEM DRIVEN
MAINTAINABILITY
• WORK CLEARANCES
• ACCESSIBILITY
• TROUBLESHOOTING
• REPLACEMENT TIMES
• REPAIR TIMES
• REDUCE POTENTIAL FOR MAINTENANCE ERROR
• PACKAGING & MOUNTING PROVISIONS
TESTABILITY
• FD/FI COVERAGE
• FAULT LATENCY
• FUNCTIONAL PARTITIONING
• PERFORMANCE MONITORING
• TEST EQUIPMENT COMPATIBILTIY
MTBF = MEAN TIME BETWEEN FAILURE
MTTR = MEAN TIME TO REPAIR
MLDT = MEAN LOGISTICS DELAY TIME
MDT = MAINTENANCE DOWNTIME
MRT = MEAN RESTORE TIME
R89-0687-014
R88-7339-018

Figure 5-1. Reliability Relationships with Maintainability & Testability.

impact of maintainability and testability design features. MLDT and MDT parameters are functions of the logistics system in place during the operations and support phase of the system life cycle. MTTR is directly influenced by design, and more specifically, by maintainability and testability. Figure 5-1 lists many of the maintainability and testability design aspects/features that impact MTTR.

The ability to meet fault tolerance requirements imposed upon a system is directly related to its capability to detect, isolate, and repair malfunctions as they occur or are anticipated to occur. This mandates that alternate maintainability diagnostic concepts be carefully reviewed for effectiveness before committing to a final design approach. A maintenance concept based upon the system's maintainability features and diagnostic capabilities must then be developed to optimize logistics resource requirements. The repair scenario should be viewed from as global a position as possible to accurately determine the potential, bottom-line impact of the fault tolerance diagnostics on LCC. Unscheduled organizational (O)-level maintenance, although a major driver of LCC, is only a portion of the total overall maintenance activity impacted by the inherent diagnostic capability. Other maintenance activities affected include scheduled/preventive, O-level inspection and service, intermediate (I)-level maintenance, and depot (D)-level maintenance. As a result, the technical manager must strive to integrate the diagnostic requirements for maintenance with those necessary to implement the fault tolerant design approach in order to properly control system effectiveness and LCC.

5.2 Maintainability Concept

The technical manager must assure that an effective maintainability and diagnostic concept is defined that is capable of meeting all of the mission performance requirements while at the same time minimizing LCC. Since there are generally a number of options available, some basic and typical questions that should be answered by a technical manager defining a fault tolerant system include:

a. What are the overall mission reliability and fault tolerance requirements and how will they impact the diagnostic requirements?

b. Do these fault tolerance requirements demand multiple redundancies and/or sophisticated techniques to enhance mission reliability?

c. What are the system functional performance monitoring requirements and can they be utilized for a fault tolerant approach?

d. An early decision will be necessary to determine whether the fault tolerant system will be "attended" or "unattended" during its normal operation. If "unattended", the design must incorporate all of the diagnostics and logic necessary to automatically recognize and eliminate malfunctions. Can the system design provide enough computer power and resources to accomplish this functional requirement effectively?

e. If the system will be "attended", the provisions for on-line maintenance should be considered. Has physical access been provided to facilitate maintenance and adjustment while the system is on-line?

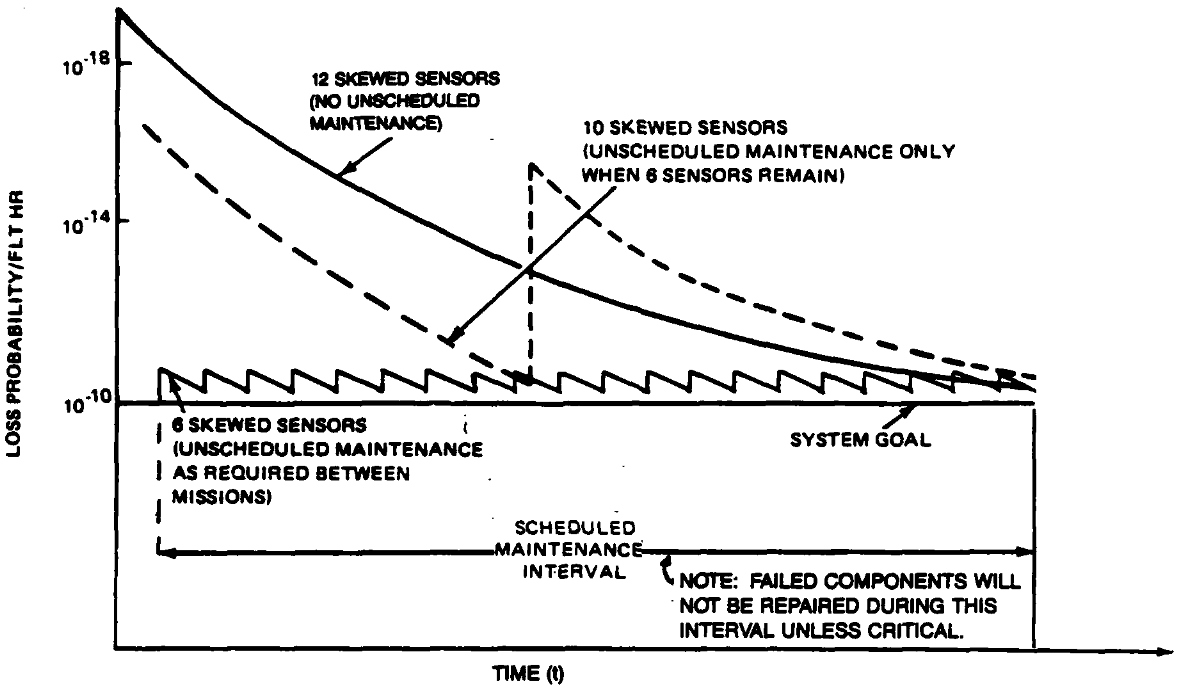
f. Can the equipment design be functionally partitioned to facilitate diagnostics and a module-level repair concept?

The appropriate answers to these and other pertinent questions will help formulate the maintainability and diagnostic concepts necessary for an effective fault tolerant system design. The maintainability and diagnostic concepts are basically design approaches that will influence and guide the design process. Given the realities of the system design and development process with its limited resources and constraints, the technical manager must assure that compromises are not made which may preclude attainment of all desirable maintainability and diagnostics goals.

A practical maintenance concept is then formulated, based on the actual design features and performance levels provided in the final design. Ideally, this maintenance concept should utilize the available capabilities of the design and should structure the scheduled and unscheduled maintenance activities to augment and complement the fault tolerant aspects of the system.

5.2.1 Design vs Corrective Maintenance Tradeoff

Figure 5-2 illustrates the design vs corrective maintenance tradeoff analysis needed early in the program phase to achieve reliability and availability goals. This figure illustrates the redundancy restoration frequency for three fault tolerant system approaches. It indicates that at some maintenance cost, the restoration frequency can be traded off against sensor redundancy levels. This particular program required a time period for allocating a scheduled maintenance activity (shown horizontally on the graph). The program also required a probability of less than one in 10 billion per flight hour that a total loss of the skewed sensor function (and therefore a catastrophic system failure) would occur.



NOTE: MORE FREQUENT RESTORATION OF REDUNDANCY LOWERS FAULT TOLERANCE REQUIREMENTS, BUT RESULTS IN HIGHER MAINTENANCE MANHOURS.

R89-0687-015
R86-2131-013
R86-7339-019

Figure 5-2. Effect of Maintenance Concept on Level of Fault Tolerance.

As indicated in the figure, this design goal can be met (in part) by incorporating either 12, 10, or 6 redundant skewed sensors. If twelve redundant skewed sensors were deployed, no unscheduled corrective maintenance would be anticipated between deferred maintenance cycles. If ten redundant skewed sensors were deployed, only one unscheduled maintenance action would be anticipated. If six redundant skewed sensors were deployed, approximately twenty unscheduled corrective maintenance actions would be anticipated between the scheduled maintenance cycles. Therefore, the decision as to how many redundant skewed sensors to use was made by first answering questions about the resultant unscheduled corrective maintenance requirements. Typical questions to answer for this type of analysis include the following:

a. What methods will be used to fault detect and fault isolate a failed skewed sensor? How effective will the FD/FI tests be? What faults cannot be detected and/or isolated using the FD/FI tests?

b. What is the risk that an unscheduled corrective maintenance action will adversely affect the mission? Is it tolerable?

c. How many manhours would be necessary to perform anticipated unscheduled maintenance actions?

d. What is the MTTR for such a system, and does the MTTR meet the system performance requirements?

e. Can the system provide full service during an unscheduled maintenance activity?

f. How many spares must be stocked and at how many locations?

g. How long does it take to replenish the spares inventory?

Figure 5-3 presents key attributes of the options available for maintaining fault tolerant C^3I systems.

5.2.2 Scheduled Maintenance & Prognostics

Fault tolerant designs, with their superior diagnostics, are particularly well suited to an "on-condition" maintenance approach. Given that the fault tolerant diagnostics are fully capable of automatically monitoring all major/critical functions for degradation as well as failure, the "on-condition" maintenance approach will be effective in reducing the costs of maintenance and the effects of unnecessary and frequent equipment removals. The close monitoring and tracking of performance degradation also facilitates the attainment of a deferred maintenance approach (see Subsection 5.2.3).

Mechanical equipment has traditionally been subject to a firm or fixed service and overhaul schedule due to refurbishment requirements or wear out conditions. Recent major improvements in monitoring and BIT capability for mechanical equipment (i.e., powerplant, hydraulic, landing gear, flight controls, etc) will permit the application of the same "on-condition" and "deferred" maintenance approach now utilized for avionics. When fault tolerant requirements are imposed on mechanical equipment design, the same prognostic capabilities (including the precise measurement and tracking of degradation) can be employed. Removals for overhaul or major service can now be safely deferred to a more convenient time or location with confidence as the degraded performance of the system is moni-

MAINTENANCE CONCEPT	DESCRIPTION	TYPICAL APPLICATIONS	ADVANTAGES	DISADVANTAGES
ON-LINE	DESIGN ALLOWS RAPID RESTORATION OF THE SYSTEM BY REPLACEMENT OF BIT IDENTIFIED LRU's AND LRM's WITH SPARES.	HIGH CRITICALITY STRATEGIC SYSTEM FUNCTIONS, i.e., DATA PROCESSING, COMMUNICATION LINKS, ETC. ALSO IN-FLIGHT ON-EQUIPMENT MAINTENANCE WITH ON-BOARD SPARES.	SYSTEM CONTINUES FULL OPERATION OR WITH MINOR INTERRUPTION IN SERVICE.	ADDED COMPLEXITY OF FD/FI, AND SWITCHING. ADDED COST OF DUPLICATED EQUIPMENT AND ON-LINE SPARES.
DEFERRED	DESIGN ALLOWS SCHEDULING NON-CRITICAL MAINTENANCE AT A MORE CONVENIENT TIME OR PLACE.	ACCEPTABLE DEGRADED MODES OF OPERATION AND OTHER GRACEFULLY DEGRADING SYSTEMS. NON-CRITICAL EQUIPMENT FAILURES.	SYSTEM CONTINUES OPERATING. MORE EFFICIENT USE OF MAINTENANCE MANPOWER AND SCHEDULE.	FULL PERFORMANCE CAPABILITY MAY NOT BE AVAILABLE, IF NEEDED.
OPPORTUNISTIC	DESIGN ALLOWS CONTINUED OPERATION WITH A DEGRADED SYSTEM UNTIL THE REQUIRED MIX OF SPARES, ATE, PERSONNEL AND SCHEDULE IS AVAILABLE TO PERFORM THE DEFERRED MAINTENANCE.	ACCEPTABLE DEGRADED MODES OF OPERATION AND OTHER GRACEFULLY DEGRADING SYSTEMS. NON-CRITICAL EQUIPMENT FAILURES.	SYSTEM MAINTAINS HIGH READINESS. MORE EFFICIENT USE OF MAINTENANCE MANPOWER AND SCHEDULE.	FULL PERFORMANCE CAPABILITY MAY NOT BE AVAILABLE, IF NEEDED.
PREPOSITIONED	COMPREHENSIVE MAINTENANCE IS LIMITED TO SPECIFIC SITES. THE SYSTEM CAN BE DIVERTED OR TRANSPORTED FROM ITS OPERATIONAL SITE TO A PARTICULAR MAINTENANCE SITE TO PERFORM A PARTICULAR LEVEL OF MAINTENANCE.	AIRBORNE C^3I SYSTEMS AND TRANSPORTABLE SUBSYSTEMS.	REDUCED MAINTENANCE MANPOWER, SKILL LEVELS AND SUPPORT EQUIPMENT REQUIRED.	MAY RESULT IN DEGRADED READINESS.
RAPID DEPLOYMENT	DESIGN PERMITS SYSTEM OPERATION FOR A SPECIFIC TIME PERIOD WITH MINIMUM LOGISTICS AND SUPPORT RESOURCES.	GROUND MOBILE AND AIRBORNE C^3I SYSTEMS WITH SELF-CONTAINED ELECTRICAL GENERATORS, AUXILIARY POWER UNITS, JET FUEL STARTERS, ETC.	ENHANCED TACTICAL/ SURGE CAPABILITY DURING HOSTILE ACTIONS.	ADDED SYSTEM COMPLEXITY.
AUSTERE SITE	DESIGN PERMITS SYSTEM OPERATION FOR EXTENDED TIME PERIODS AT UNIMPROVED FACILITIES WITH MINIMAL LOGISTICS RESOURCES.	GROUND AND AIRBORNE C^3I SYSTEMS WITH SELF-CONTAINED ELECTRICAL GENERATORS, AUXILIARY POWER UNITS, JET FUEL STARTERS, ETC.	ENHANCED SYSTEM SURVIVABILITY DURING HOSTILE ACTIONS.	ADDED INITIAL SYSTEM COST.
SELF CONTAINED	A SYSTEM CONTAINING SUFFICIENT FAULT TOLERANT DESIGN PROVISIONS THAT REQUIRES LITTLE OR NO EXTERNAL MAINTENANCE TO COMPLETE A MISSION.	HIGH CRITICALITY STRATEGIC NATIONAL SECURITY SYSTEMS.	HIGH READINESS.	ADDED COMPLEXITY, WEIGHT, POWER AND INITIAL COST.

R80-0867-016
R87-3637-016(T)
R86-7339-080

Figure 5-3. Maintenance Concept Options.

tored. Performance excursions beyond safe limits will require the safe shutdown and by-pass of the failed function or item in the system.

5.2.3 Deferred Maintenance Approach

Due to the high cost of C^3I equipment maintenance and the limited availability of adequately skilled personnel, there is increasing emphasis on providing fault tolerance to keep a system running until maintenance can be performed, or at least until maintenance personnel are available to make needed repairs. This concept, coupled with the requirement for continuity of service in most military systems, usually results in added redundant components. Even though the redundant equipment adds to the overall maintenance burden, there is an advantage in that full service can be maintained by operating back-up (redundant) units while maintenance is performed on the failed units.

Specific examples of "hot" maintenance techniques would include: a central computer or processor function composed of 3 or 4 redundant computers where self-contained diagnostics identify the failed computer and assign its processing load to the remaining good computers. Another example would be a fly-by-wire flight control system where triplex redundant servo control valves are employed. When one channel is found faulty by system and unit built-in-test, it can be shut down and control passed to the remaining "good" channels. This would permit "deferred" or "opportunistic" maintenance to be effectively utilized. A complex VLSI type chip with four redundant functional channels and self-contained diagnostics could detect and isolate a defective channel without loss of system functionality. Again, "deferred" or "opportunistic" maintenance could be effectively utilized to preserve the fault tolerant features of the system.

When "opportunistic" and "hot" maintenance is to be performed on these fault tolerant systems, special care must be exercised in the design to assure safe maintenance (free of hazards such as exposure to thermally hot surfaces or liquids, high pressure, radiation, electrical shocks, or moving equipment). In fact, the safest approach might be to partition the system such that the channel, section, subsystem, or equipment requiring maintenance can be shut down while the overall system is still functioning.

Technical managers should consider the following when reviewing deferred maintenance practices:

a. Are maintenance technicians and tools available at the appropriate time and location?

b. Is adequate accessibility designed into the system?

c. Can maintenance be performed without interfering with the operating units?

d. Are backup power supplies required?

e. Are degraded modes of operation required and/or acceptable?

f. Are there opportunities for modularity?

g. Can common support equipment be used?

h. Is there any special design consideration for maintenance in the operational environment (while the equipment is functioning; for potentially hostile environments)?

All C^3I systems, no matter what maintenance concept option is chosen, will require a high level of maintainability in the design to minimize the time, effort, and cost of performing maintenance. This is especially important with the increased complexity of fault tolerant systems. Because of this complexity, C^3I technical managers and designers must emphasize maintainability and ensure that it is incorporated into the equipment design.

5.2.4 General Maintainability Considerations

There are some unique maintainability considerations that are particularly applicable to the design of fault tolerant systems. Given that the system requirement includes near 100% functional operational capability at all times or throughout an identified period or mission, the following maintainability concerns must be addressed:

- Have the necessary redundancy and diagnostic requirements been defined to carry out the functional fault tolerant mandate?
- Will the system be unattended (i.e., must it function without operator intervention)? If so, it must have the self-contained logic and spare functions to achieve fault tolerant performance requirements throughout its mission.
- Has the system been designed to permit easy access to repairable

or replaceable elements with minimal use of special tools and support equipment?

- Has the system been functionally modularized to facilitate fault localization and easy replacement of failed items?
- Has the system equipment been designed to permit replacement and sparing of the smallest practical functional module?
- Can the fault localization diagnostics isolate to the faulty module 100% of the time?
- Have equipment and replaceable element mounting provisions been designed to eliminate or minimize the need for attachment hardware and special tools?
- Has the system design and partitioning concept considered state-of-the-art approaches such as integrated racks and small, plug-in modules?
- Has the concept of reconfiguration management and resource sharing been incorporated into the overall maintainability approach? Have the necessary diagnostics and logic been defined to properly control and maintain the system functions during periods when failures occur and/or maintenance is being performed?

5.2.5 Maintainability Design Criteria

In order to translate maintainability requirements and anticipated operational constraints into practical and effective fault tolerant hardware designs, a broad spectrum of maintainability design criteria, both general and specific, must be defined and employed during the early phases of system design. Design criteria may be in the form of requirements and/or guidelines. Requirements are usually derived from contractural documents and are contained in specifications such as the Prime Item Development Specification. Quantitative maintainability requirements and any special requirements such as diagnostics, BIT, and testability are defined in these specifications. Guidelines are usually qualitative in nature and provide a recommended course of action to achieve the maintainability design objectives and goals. General maintainability design guidelines have been defined in MIL-STD-470 and AF DH-1-9 to assist the technical manager in formulating a maintainability design strategy. MIL-STD-2084 and MIL-STD-2165 provide insight into avionics and testability designs for maintainability.

5.2.6 Accessibility

A prime consideration in maintainability design is accessibility which directly affects the elapsed time to repair and the readiness of the system. An accessible component is easier to maintain and can significantly increase the efficiency of the technician working on the equipment. Simplicity is the key in designing for accessibility. Simplification minimizes the number of parts, interconnections, and fasteners, minimizes the number and simplifies the design of support tools/equipment, and makes components and test points easily accessible. Techniques for improving accessibility must be implemented early in the design process; hence, early involvement in the design process is an integral part of proper maintainability design.

The following subsections discuss factors with which the technical manager should be concerned when designing equipment for improved accessibility.

5.2.6.1 Equipment Access & Fasteners

Figure 5-4 lists recommended design methods for equipment access, in preferential order. Accessibility is directly affected by the number and type of fasteners used to secure access doors and maintainable items. The fewer the fasteners to be removed or released and the easier such fasteners can be removed or released, the lower the required maintenance time. Figure 5-5 summarizes the desirability of various fastener types.

5.2.6.2 Packaging & Connectors

Within the context of fault tolerant hardware design and construction, packaging encompasses the physical methods used to assemble electronic equipment. Packaging affects maintainability down to the lowest-level throwaway item and obviously impacts repair time and diagnostics. Figure 5-6 discusses the desirability of various packaging methods. The word "item" in this figure represents a replaceable item or an assembly of such items (such as a repairable module), or an assembly of such assemblies, and so on, up to the system level. Quickly replaceable mother boards and wiring harness assemblies are also to be considered as "items" for fast repair under a fault tolerant design approach.

DESIRABILITY	FOR PHYSICAL ACCESS	FOR VISUAL INSPECTION ONLY	FOR TEST AND SERVICE EQUIPMENT
MOST DESIRABLE	INTEGRATED PLUG-IN RACK CONCEPT	OPENING WITH NO COVER (IF PRACTICAL) OR PLASTIC WINDOW OR QUICK OPENING HINGED DOOR	OPENING WITH NO COVER (IF PRACTICAL) OR QUICK OPENING HINGED DOOR
MORE DESIRABLE	PULL-OUT SHELVES OR DRAWERS	OPENING WITH NO COVER OR PLASTIC WINDOW	OPENING WITH NO COVER (IF PRACTICAL) OR QUICK OPENING HINGED DOOR
DESIRABLE	QUICK OPENING HINGED DOOR (IF DIRT, MOISTURE, OR OTHER FOREIGN MATERIALS MUST BE KEPT OUT)	PLASTIC WINDOW (IF DIRT, MOISTURE, OR OTHER FOREIGN MATERIALS MUST BE KEPT OUT)	SPRING-LOADED SLIDING CAP (IF DIRT, MOISTURE, OR OTHER FOREIGN MATERIALS MUST BE KEPT OUT)
LESS DESIRABLE	REMOVABLE PANEL WITH CAPTIVE, QUICK-OPENING FASTENERS (IF THERE IS NOT ENOUGH ROOM FOR HINGED DOOR)	BREAK-RESISTANT GLASS (IF PLASTIC WILL NOT STAND UP UNDER PHYSICAL WEAR OR CONTACT WITH SOLVENTS)	HINGED ACCESS DOOR
LEAST DESIRABLE R89-0687-017 R88-7339-022	REMOVABLE PANEL WITH SMALLEST NUMBER OF LARGE SCREWS THAT WILL MEET REQUIREMENTS (IF NEEDED FOR STRESS, PRESSURE, OR SAFETY REASONS)	COVER PLATE WITH SMALLEST NUMBER OF LARGE SCREWS THAT WILL MEET REQUIREMENTS (IF NEEDED FOR STRESS, PRESSURE, OR SAFETY REASONS)	COVER PLATE WITH SMALLEST NUMBER OF LARGEST SCREWS THAT WILL MEET REQUIREMENTS (IF NEEDED FOR STRESS, PRESSURE, OR SAFETY REASONS)

Figure 5-4. Recommended Equipment Access Provisions.

DESIRABILITY	TYPE OF FASTENER
MOST DESIRABLE	QUICK RELEASE, CAPTIVE
DESIRABLE	COARSE-THREAD SCREWS WITH CAPTIVE NUTS
	FINE-THREAD SCREWS WITH CAPTIVE NUTS
LESS DESIRABLE	SCREW AND NON-CAPTIVE NUT
LEAST DESIRABLE	RIVET OR EYELET
R89-0687-018 R88-7339-023	

Figure 5-5. Recommended Fastener Types for Accessibility.

DESIRABILITY	METHOD OF PACKAGING	
	MOUNTING	INTERCONNECTION
MOST DESIRABLE	(1) ITEMS PLUGGED INTO SOCKETS ON SUPPORTING MEMBER (MOTHER BOARD, CHASSIS, CIRCUIT BOARD, ETC.) AND CLAMPED DOWN	(1) PRINTED CIRCUITS POINT-TO-POINT WIRING, CABLING, ETC. SOLDERED (OR EQUIVALENT) TO SOCKET
DESIRABLE	(2) ITEMS FASTENED (SCREWS, ETC.) TO SUPPORTING MEMBER; CONTACTS COMPLETED BY ATTACHING A CONNECTOR OR CONNECTORS	(2) SAME AS (1), EXCEPT TO CONNECTOR(S) INSTEAD OF SOCKETS
LEAST DESIRABLE	(3) SAME AS (2), EXCEPT THAT CONTACTS ARE COMPLETED BY ATTACHING INDIVIDUAL LEAD TO EACH CONTACT OR TERMINAL STRIP	(3) SAME AS (1), EXCEPT TO ITEM CONTACTS INSTEAD OF SOCKET

R89-0687-019
R88-7339-024

Figure 5-6. Desirability of Packaging Methods.

Many types of connectors are available to equipment designers. The major factors to be considered in selecting connectors for maintainability are repairability, connector size, space available, method of insertion and removal, forces required for insertion and removal, method of securing the connector, polarization methods, and the means used to connect the inter-connecting wiring to the connector. In addition, connectors must be designed to prevent coupling misalignment and inadvertent interchange (i.e., Murphy proofing).

5.2.6.3 Test Point Accessibility

Fault tolerant system designs should aim for an embedded diagnostic capability that will automatically detect, locate, and isolate faults not only to the faulty function for reconfiguration purposes, but to the replaceable module level for maintenance purposes. As a result of imperfect fault isolation, effective system level test points and their accessibility become crucial requirements if rapid repair actions are necessary to provide and maintain system operation and/or system fault tolerant capability. **Failures in wires, connectors, harnesses, mother boards, and other interconnection devices are often difficult to isolate but can be handled by the proper design and use of test points in critical areas (signal paths)**

where wrap-around tests or signal insertion/pick-off methods are utilized in combination with BIT and self test capabilities. If external monitoring and control is needed to provide the required rapid fault isolation capability, the "test points" must be made readily accessible to the maintainer. The maintainer should be equipped with automatic or semi-automatic external test equipment that can be quickly attached to these external test points. System level test points should be provided as centrally located connectors or "ports".

The accessibility of test points at lower equipment levels may be a major concern when the use of external test equipment is part of the selected maintenance plan. Built-in-test equipment ordinarily is connected permanently to the appropriate system test locations and the interface of test functions can be a problem. Maximum accessibility is achieved when all test points are brought to the outside of the smaller items being tested. If manual testing methods are employed, logical groupings (from the viewpoint of signal flow) and clear markings are also required for the best accessibility. When automatic and semiautomatic testing methods are used, the test points should interface with the tester through a minimum number of multiple-contact connectors located on the face of the item, rather than through large numbers of individual test points. Less preferable, but acceptable, are internal test points (for manual testing) located close to the circuit elements for which they serve as input or output points. These locations should be easily accessible, and system operation should not be interrupted to engage the test point. Similarly, connectors to be used with automatic and semiautomatic testers may also be located inside the item, under the same conditions of accessibility and operability. Figure 5-7 summarizes test point accessibility considerations.

5.2.7 Maintenance Personnel

The Air Force has made significant strides in incorporating new technologies to increase the capability of new and existing C^3I systems. However, it is important that these new technologies do not require field maintenance personnel to undergo retraining, or develop higher skill levels. A major maintainability objective for new C^3I systems should be the reduction in manpower requirements per system operating hour. Systems

RATING	MANUAL TESTING	AUTOMATIC AND SEMIAUTOMATIC TESTING
PREFERRED	LOCATED ON FACE OF ITEM IN LOGICAL GROUPINGS (SIGNAL FLOW); CLEAR MARKING; INDIVIDUAL TEST POINTS	LOCATION ON FACE OF ITEM; MULTIPLE-CONTACT CONNECTORS, AS FEW AS POSSIBLE; CLEAR MARKING
ACCEPTABLE	LOCATED CLOSE TO ELEMENT WITH WHICH ASSOCIATED, IF ACCESSIBLE WHILE ELEMENT IS OPERATING; CLEAR MARKING; INDIVIDUAL TEST POINTS	INTERNAL LOCATIONS IF ACCESSIBLE WHILE EQUIPMENT IS OPERATING; CLEAR MARKING; MULTIPLE-CONTACT CONNECTORS, AS FEW AS POSSIBLE
R89-0687-020 R88-7339-028		

Figure 5-7. Test Point Accessibility Considerations.

must be designed to allow technicians to easily and accurately diagnose and repair a wide variety of system and subsystem failures. The approach calls for more standard operating, maintenance, and testing characteristics across similar subsystems and more emphasis on maintainability in the design process. This approach will reduce the multiple specialties required to service increasingly complex systems.

Significant reductions in maintenance manpower requirements depend on how well technical managers plan and design their fault tolerant systems and manage new technologies. Designing with the maintenance technician in mind must be an integral part of the system design process. System designers must be encouraged to standardize maintenance functions across subsystems (e.g., electronic computers, etc). All new C^3I systems should be designed with functional modules that have simple and standard fault identification procedures that are common to all like subsystems regardless of the component, its function, technology, or application. On-equipment maintenance actions should be confined to fault detection, isolation, removal, and replacement. Built-in diagnostics should allow field-level replacement of failed units without using external test equipment.

Standardization and simplification of the equipment design will eliminate the need for certain specialties and enable the consolidation of specialties with similar skill requirements. For example, an "electronics" skill specialist can maintain radar, communications sets, electronic warfare systems, etc, that were designed with modular components and with dependable and repeatable fault isolation to the LRU or circuit card level. The electronics technology of C^3I systems offers the most immediate potential for implementing simple, reliable fault isolation to the lowest field replaceable module. Electronic and electromechanical replaceable units should have built-in-test/fault-isolation-test indicators on the unit that will indicate when a fault occurs and will hold that indication until a repair is made.

The added complexity (i.e., active redundancy, hot/cold standby spares, voting schemes, etc) inherent in fault tolerant systems requires the development of improved maintenance procedures and manuals that will ultimately ease the maintenance tasks and reduce maintenance downtime. Some considerations for improving the text and content of maintenance manuals include:

- Components that are functionally dependent should be grouped and identified with more consideration given to information flow or circuit configuration
- Controls, indicators, test locations, etc, should be clearly labeled and identified
- Relationship between circuitry, functions, and hardware boundaries should be clearly indicated
- Written text should be kept to a minimum, used only for essential explanation, and presented in a style that is easily understood by the maintenance technician.

5.2.8 Maintainability Checklist Questions

a. Will the maintainability concepts be developed in parallel with other concepts proposed for achieving reliability, availability, and survivability requirements?

b. Have the life cycle costs of various maintainability design options been considered before establishing a maintenance concept?

c. Which maintainability design options will best provide an efficient and cost-effective means to maintain a C^3I system v.:thout hindering mission performance?

d. Will deferred or opportunistic maintenance be a potential policy, and if so, have all the issues involving system design as well as

the impact on maintenance technicians, tools, and operational scenarios been considered?

e. How much downtime is allowed and how will it relate to the system's availability?

f. Once the maintenance concept is developed, are all the factors in the maintainability concept being applied to the design of the equipment?

g. Has a set of suitable maintainability design criteria been established for the prcgram?

h. Have inherent maintainability capabilities in the equipment design been achieved by careful consideration and optimum balance among the following factors:

   - Basic physical configuration and layout of the design for quick and easy access for maintenance
   - Test provisions for fast and accurate fault isolation to the replaceable item level
   - Use of methods for quick disconnection, reconnection, and hold-down of replaceable items
   - Interchangeability of replaceable items for minimum adjustment and alignment during or following replacement
   - Provisions for rapid post-maintenance checkout to verify restoration to specified performance levels
   - Utilization of standard test equipment and tools for maintenance
   - Adequacy, clarity, and simplicity of maintenance procedures, instructions, and documentation
   - Compatibility of available skill levels and technician training with maintenance tasks unique to the design

i. Has the design approach for maintenance and testing been standardized and simplified and has this effort been implemented early in the design process?

5.3 Testability of Fault Tolerant Designs

An essential element of fault tolerance and system level reliability is accurate diagnostics, including the ability to detect and isolate faults in redundant elements. For these reasons basic design concepts for testability and, in particular, built-in-test schemes must be considered early in the concept definition phase of fault tolerant system development (Ref 17).

For fault tolerant systems, it is important that the design's inherent testability provisions include the ability to detect, identify, recover, and if necessary reconfigure, and report equipment malfunctions to operational personnel. In addition, because fault tolerant systems often are characterized by complex, non-serial reliability block diagrams, a multitude of backups with non-zero switchover times, and imperfect fault detection, isolation, and recovery, it is imperative that the technical manager assure that effective testability provisions are incorporated in the system design concept. If not, the design, when fielded will exhibit long troubleshooting times, high false alarm rates, and low levels of system readiness.

The terms *integrated diagnostics, diagnostics,* and *testability* are often used interchangeably when describing the test capability of a system. Since the technical manager must understand the basic differences in the use of these terms, a brief discussion of each is provided in the following paragraphs.

Integrated diagnostics is the *process* that translates system performance, mission, and mobility objectives into test performance capabilities. The goal is to detect and isolate all faults to the replaceable item, with a minimum of false removals and unnecessary maintenance actions, using a mix of test capability both internal (i.e., built into the system) and external to the system. The mix that makes up the test capability is selected from design techniques (e.g., built-in-test, status monitoring, partitioning, test points), external hardware and software (e.g., automatic and manual test equipment), technical information (e.g., technical manuals, information systems, operator displays), and maintenance personnel attributes (e.g., skill levels, training). The maintenance levels considered should include organizational, intermediate, and depot if the system employs a 3-level maintenance concept. Reference 18 addresses the integration (during the system acquisition process) of the on-equipment and off-equipment test resources needed to provide an integrated diagnostics capability. In addition, the Air Force Generic Integrated

Maintenance and Diagnostic System, currently under development, provides a roadmap for the development of integrated diagnostics. **The technical manager should be aware that the current trend in planning the maintenance level activity for new system acquisitions is to minimize, or if possible eliminate, intermediate level maintenance requirements.** This trend is being fostered by the high cost and mobility requirements associated with intermediate level maintenance activity experienced on recent deployments.

Testability is defined in MIL-STD-2165 as a *design characteristic* which allows the status (operable, inoperable, or degraded) of a system/-item to be determined and enables the isolation of faults within the system/item in a timely manner. Testability establishes the test capability built into the system/item and impacts, via its designed-in interfaces (i.e., test points, displays, etc), the test capability external to the system (e.g., automatic and manual test equipment, technical information, maintenance technician training levels, etc). The tasks for establishing a testability program are fully described in MIL-STD-2165. The tasks and the tailoring of the testability program requirements are discussed in Subsection 3.3 and Ref 32.

The third term, *diagnostics*, usually refers to the subset of testability which is concerned with the functional localization of a fault or the physical isolation of the fault to a replaceable item.

To implement testability into the system, the technical manager must ensure that the testability program that is developed is compatible with the integrated diagnostics concept that has been defined for the system and its interfaces. In addition, it is important that the technical manager have an understanding of the testing options which are available when establishing testability design criteria. These testing options are discussed in the following subsections.

5.3.1 Testability Concepts

Testability is an integral subset of the integrated diagnostic capability of the system and is primarily concerned with the inherent test capability built into the system and its designed-in interfaces. The technical manager must understand the implications of the system's design, planned operational, deployment and maintenance concepts on testability design criteria. The system's architecture, mission reliability, sortie rate, turnaround time, mobility requirements, planned levels of maintenance (e.g., organizational, intermediate, depot) and planned maintenance resources (e.g., test equipment, maintenance personnel) have a direct influence on the choice of the testing options.

Figure 5-8 illustrates the trend in system architecture toward distributed systems with digital implementation. In early systems, information was displayed to the operator who functioned as the system integrator and the focal point for system health and status information. Most modern systems utilize digital computers to integrate and distribute data; future systems will consist of highly distributed hardware and software that will have high levels of system integration. These sophisticated systems with high levels of integration have the potential to significantly increase system/operator effectiveness; however, these systems require complex testing and troubleshooting procedures which might adversely impact system readiness and LCC unless adequate and effective testability design provisions are included in the system design.

Figure 5-9 identifies the testability options available through either manual or automatic test techniques. These test techniques can be applied individually or in combination at any level of the system design (e.g., system, subsystem, equipment, module, component) to aid in fault detection and isolation to that level. They can also be applied individually, or in combination, to enhance testing at any of the maintenance levels (e.g., organizational, intermediate, depot). The trend to minimize or eliminate intermediate maintenance activity requires the technical manager to place more emphasis on the implementation of testability techniques which would be utilized at the organizational maintenance level.

5.3.1.1 Manual Test

The manual test concept relies completely on manual operation, operator decision, and operator evaluation of test results. Testability is im-

1940 - 60 ANALOG	1960 - 80 DIGITAL COMPUTERS	1980 - 2000 DISTRIBUTED DIGITAL
• LITTLE FAULT TOLERANCE	• SOME FAULT TOLERANCE	• FAULT TOLERANCE
• NO REDUNDANCY	• SOME REDUNDANCY	• LARGE SCALE USE OF REDUNDANCY
• NO DYNAMIC RECONFIGURATION CAPABILITY	• NO DYNAMIC RECONFIGURATION CAPABILITY	• DYANMIC RECONFIGURATION
• DISCRETE AND SSI HARDWARE	• MSI AND LSI HARDWARE	• VHSIC HARDWARE
• WIRED PROGRAMS	• STORED PROGRAM	• DISTRIBUTED HIERARCHICAL STORED PROGRAM
• DEDICATED ANALOG PROCESSORS	• PROCESSOR(S)	• REDUNDANT PROCESSOR(S)
• INTEGRATION THROUGH PILOT/DISPLAYS	• COMMUNICATION THRU I/O INTEGRATION THROUGH PROCESSOR/STORED PROGRAM	• DISTRIBUTED DEDICATED FUNCTIONAL PROCESSORS
		• COMMUNICATION THRU BUS NETWORK

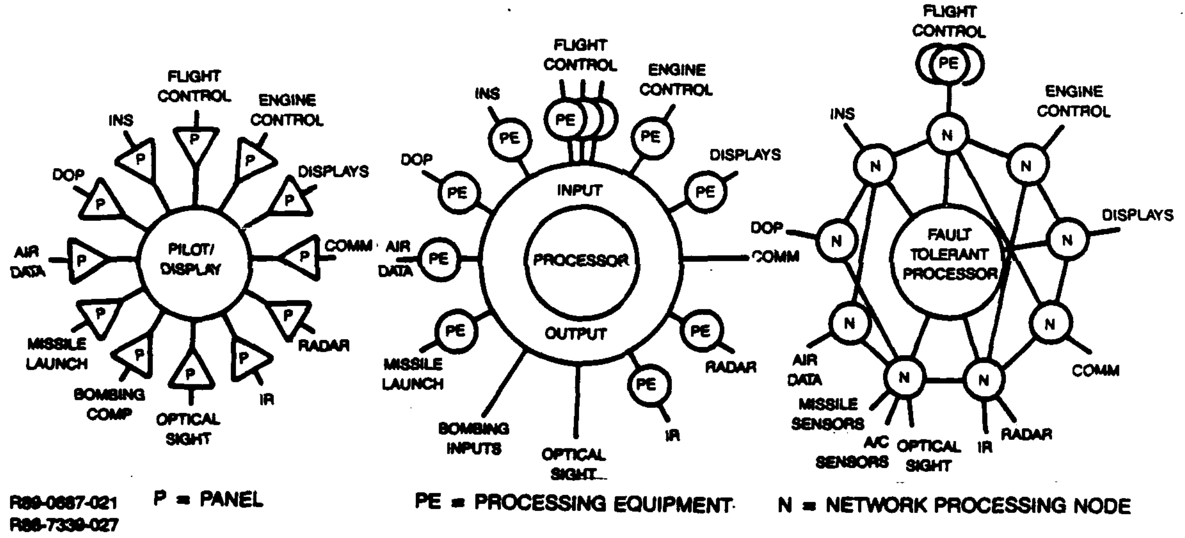


Figure 5-8. System Technology Trends Impacting Testability.

plemented by providing designed-in test points that interface with test equipment. The test equipment may be a specially designed physical part of the system, or it may consist of standard off-the-shelf test equipment (i.e., oscilloscopes, signal generators, voltmeters, counters, etc) which can be attached to designated external test points. The technical manager must recognize the following essential characteristics of the manual test concept as applied to fault tolerant system designs:

- The test equipment is manually operated
- Test results are evaluated by the operator/maintenance technician

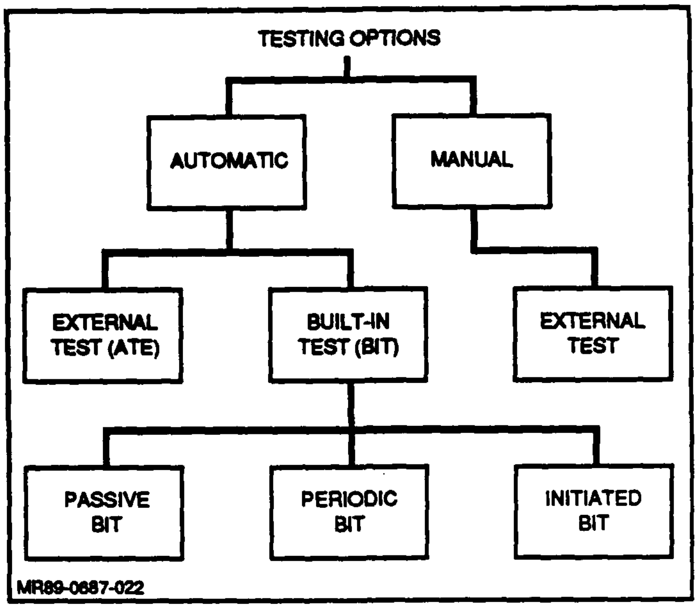


Figure 5-9. Testing Options for Testability Design.

- The interface with the operating system is usually through designed-in (designated) test points which require functional buffering and physical accessibility
- Any switchover to redundant equipment is performed manually.

5.3.1.2 Automatic Test

In the automatic test concept, testability features are designed to be automatic rather than manual. Under this concept, performance assessment, fault detection, diagnosis, isolation and prognosis is performed with minimum reliance on human intervention. Automatic test includes both external test and built-in-test. The automatic external test concept is similar to the manual test concept in that it usually refers to testing which is performed using a removable, stand-alone piece of automatic test equipment (ATE) that is physically separate from the system. The implementation of testability for the automatic external test concept requires designed-in test points that interface with the ATE. The major difference from the manual test concept is in the method in which the tests are initiated and the test results evaluated. The characteristics of the automatic external test concept include:

- The ATE requires minimum operator intervention

- The ATE requires minimum operator interpretation in evaluating test results
- The interface with the system is usually through designed-in (designated) test points requiring functional buffering and physical accessibility
- Switchover to redundant equipment is performed manually.

5.3.1.3 Built-In-Test

For the built-in-test (BIT) concept, the test functions are an integral part of the system design and operational hardware is made to serve a dual purpose in that it also performs test functions. BIT refers to an integral capability of the system to provide on-equipment automated test capability to detect, diagnose, and isolate equipment failures. The fault detection and isolation capability is used for periodic or continuous monitoring of a system's operational health, and for observation and, possibly, diagnosis as a prelude to maintenance. This concept is extremely important in the development of system reconfiguration strategies and fault tolerant designs. For application in fault tolerant systems, BIT must:

- Maintain the real-time status of the system's assets (both on-line and off-line equipment)
- Provide the operator with status of available system assets
- Maintain a record of hardware faults and reconfiguration events required for system recovery during the mission for post-mission evaluation and corrective maintenance.

The BIT concept may be implemented in the system in various ways. Figure 5-9 illustrates three subdivisions of the built-in-test concept which are based on the performance of built-in-test in relation to the system operational timeline. *Passive BIT* is monitoring or testing that does not disrupt or interfere with the prime system timeline. *Periodic BIT* is initiated at some predetermined frequency or within an allowable window in the prime system operational timeline. *Initiated BIT* requires operator intervention upon which the system is diverted from its operational program and dedicated to the performance of BIT. *Turn-on* or *power-up BIT* is a typical example of the *initiated BIT* category.

The essential characteristic of BIT is that it is an integral part of the system. Current fielded systems use 1% to 10% of the system hardware to test 90% to 100% of the system functions (Ref 36). In newer system designs, which rely heavily on digital technology, BIT is implemented primarily through software techniques and the hardware penalty has been reduced to less than 4% when testing 100% of the system functions. Reference 37 provides a detailed discussion of the hardware/software tradeoffs involved in establishing system test requirements. In addition, the technical manager will find additional information regarding BIT concepts, analysis techniques, and design methodologies in Ref 36.

5.3.2 Testability Design

The following subsections provide the technical manager with information on testability design for fault tolerant systems. Included is a discussion of fault tolerant design implementation and its impact on the selection of a testability approach, testability design considerations, testability design guidelines, testing in the presence of faults, fault latency times, and partitioning/levels of fault isolation.

5.3.2.1 Fault Tolerant Design Impact on Testability

Fault tolerance and recovery strategies will have a significant impact on the degree to which testability is designed into the system. For example, when incorporating testability/diagnostic capability into the design, the penalties imposed by a fault tolerant system design which employs active redundancy and voting logic may be less than those imposed by a design employing standby redundancy. With active redundancy, the prime system hardware and software are more readily adaptable to perform multiple functions (including those required for testability). In active redundant systems with voting logic, the performance/status-monitoring function assures the operator that the equipment is working properly. However, this approach also simplifies the isolation of faults since the failure is easily isolated to the locked out branch by the voting logic. In systems employing standby redundancy, test capability and diagnostic functions must be designed into each redundant or substitute functional path (both on-line and off-line) in order to determine their status.

For system designs employing active redundancy the testing options are limited to a built-in-test concept with passive BIT (i.e., continuous monitoring). Depending on the criticality of the function being tested, periodic BIT may also be used to supplement passive BIT for this type of fault tolerant design. For systems employing standby redundancy, the complete range of manual and automatic testing options shown in Fig. 5-9 are valid. The technical manager should be aware of the impact of selecting a particular fault tolerance and recovery concept on the testability of the system. The selection of a testing option, or combination of testing options, should be determined through consideration of:

- Mission length
- Allowable system downtime
- Allowable system reconfiguration (switchover) time
- Technician skill level requirements.

5.3.2.2 Testability Design Considerations

The primary objective for testability is to provide a test capability to achieve a 100% fault detection and isolation goal. For fault tolerant systems this capability provides system/equipment health/status information necessary for system reconfiguration and mission decisions (e.g., complete primary/alternate mission or abort). It also provides sufficient failure information to allow efficient and effective maintenance to be performed. **The goal of providing 100% (goal) fault detection coverage is difficult to achieve, requires interdisciplinary cooperation, the appropriate mix of hardware, test points, software, training, and technical documentation. Testability personnel should assume a lead role in developing fault detection/fault isolation (FD/FI) criteria for alternate approaches to fault tolerant system design.** Section 200 of MIL-STD-2165 describes the methodology for preliminary and detail design analyses to implement testability in the design. Reference 39 lists automated tools, currently available or in development, which would aid the technical manager in assessing the testability of the system design. Analyses performed either manually or with these automatic tools will determine:

- Fraction of faults detected
- Fraction of faults isolated

- Ambiguity group size resulting from the system's isolation capability.

The technical manager should use the results of these analyses to identify shortcomings in the fault protection coverage of the design, and establish necessary corrective actions to resolve any deficiencies.

Although the addition of redundancy is usually effective in improving system reliability, the technical manager is cautioned that the reliability improvement may be highly dependent on achievable FD/FI levels. Figure 5-10 illustrates an example where imperfect FD/FI actually causes system reliability to degrade as more redundant equipment is added. This example is based upon a subsystem composed of skewed inertial sensors which are required to stabilize a hypothetical C^3I system airborne platform. Since the sensors are skewed, a minimum of four are required to provide inertial data and meet a system loss probability goal of 10^{-10} per flight hour. Configurations of 9, 11, and 13 sensors were evaluated for compliance with this requirement. Since fault detection and fault isolation

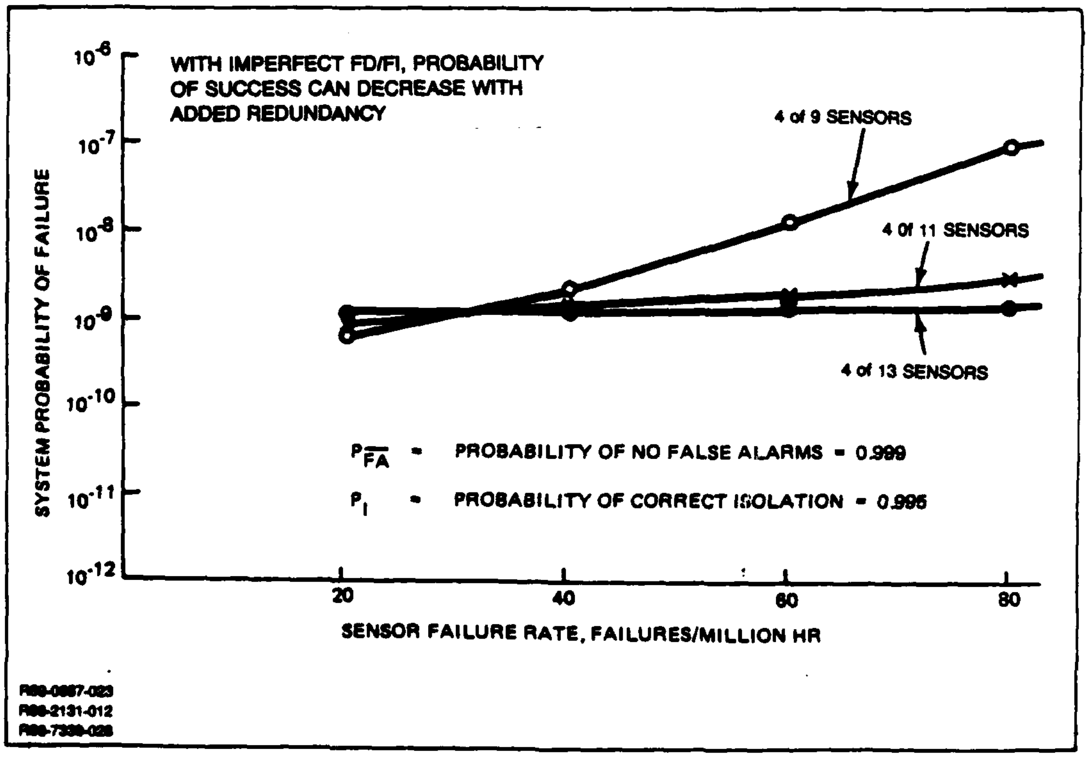


Figure 5-10. Testability Impact on System Reliability.

are an integral part of the redundancy management scheme for the skewed sensor subsystem, the analyses assumed an achievable 0.999 probability of no false alarm and a 0.995 probability of correct fault isolation. For sensor failure rates less than 30 failures per million hours, the system reliability of the 9-sensor configuration is better than both the 11 and 13 sensor configurations.

In general, the effect that varying levels of fault protection coverage have on system reliability can be evaluated by parameteric analyses. The range of fault protection coverage values used in the analyses should be based on past experience with similar hardware/software systems and adjusted by evolutionary trends and expectations for state-of-the-art devices and designs.

Additional hardware and/or software may be required to provide a test function or to provide a test interface to external equipment. As a general rule, technical managers should establish a goal that the reliability of the test circuitry which is being added should be an order of magnitude higher than the functional circuitry being tested. This assessment is made by utilizing MIL-HDBK-217 to determine the ratio of the reliability of the test circuitry components to the reliability of the functional circuitry components. This goal may be modified in the design tradeoff process if the technical manager is satisfied that it would compromise the ability to satisfy other critical system design requirements. Technical managers should assure that the ratio of test circuitry failure rate to functional circuitry failure rate in any given design is not excessive.

The following new or improved technology developments should be considered during the design and development of fault tolerant systems. The application of these developments would contribute to the achievement of the 100% FD/FI testability design goal (Ref 19):

- Very Large Scale Integration (VLSI) and Very High Speed Integrated Circuit (VHSIC) Technology - The order of magnitude reductions in the size of electronic circuits resulting from developments in advanced integrated circuit technology allows more test capability to be included in the design with negligible weight/power/volume penalties

- **Artificial Intelligence** - Artificial intelligence is one of the newer techniques which are being applied to both BIT and ATE designs (Ref 40). For fault tolerant systems this technology should be applied to functions that maintain information on system assets, and perform tests and diagnostics on off-line assets
- **Smart BIT Techniques** - Smart BIT is a product of the application of artificial intelligence techniques to specific testability problems that can not be solved by conventional techniques (Ref 20). The object of Smart BIT is to reduce the number of no-fault-found maintenance actions (i.e., Can-Not-Duplicate and ReTest OKs) through the identification of false or intermittent BIT reports. A system designed with Smart BIT can result in a substantial reduction in BIT false alarms
- **Software** - Software is an important element in the development of testability capability. Proper attention to the development of the test and diagnostic software is effective in resolving ambiguous faults and reducing ambiguity group size
- **Automated Tools** - Computer aided design techniques are available to aid in the incorporation of testability and the assessment of the testability capability during the design process. A number of tools and analysis techniques available for assessing integrated diagnostics are discussed in Ref 39. Testability features must be added early in the design and periodically evaluated to determine if testability design goals are being achieved
- **Automatic Test Equipment (ATE)** - ATE that is designed for a particular system can incorporate techniques which prompt technicians on how to set up the test interface, where to insert test probes, and how to perform specific test tasks. Such techniques have proven especially useful to break ambiguities between equipments
- **Technical Manuals** - Technicians typically require maintenance aids and documentation. Technical manuals should contain step-by-step FD/FI procedures and complete diagnostic flow charts and logic trees. Several development efforts have been undertaken

by the Air Force for the development of automated technical manuals and portable maintenance aids for the technician. The Integrated Maintenance Information System, currently under development by the Air Force Human Engineering Laboratory, will provide the technician with a global data base from which he can extract current FD/FI information for utilization with a portable maintenance aid to accomplish organizational level maintenance

- **Training** - Since the performance of effective test and diagnostics may require the intervention of the technician to set up the test equipment, initiate or perform the test, and assess/interpret test results, it is important that maintenance technicians be trained in the required test and diagnostic techniques and procedures.

When performing tradeoffs to incorporate testability features which require additional space or circuit complexity, the designer of fault tolerant systems has more freedom than the designer of conventional systems. The added hardware and software required for the testability function usually serves multiple purposes. For example, performance-monitoring (i.e., passive BIT) assures the user that the equipment is working properly and helps isolate faults to the replaceable element. In standby redundant and other configuration management strategies, the BIT or diagnostic function must detect and identify malfunctions so that the standby redundant or substitute function can be switched in. This functional requirement demands that technical managers ensure that the designers be more responsive to testability requirements and goals.

5.3.2.3 Testability Design Techniques

A number of testability techniques can be applied to fault tolerant system design. These techniques involve both on-line and off-line operational test modes. The on-line test mode may perform continuous monitoring of critical system functions and/or periodic sampling of specific system functions where the normal system is not interrupted during the test. On-line testing may also be integrated into the operating system by making use of available system dead time. On-line testing can provide immediate detection of critical system malfunctions and limited fault isolation to allow for system recovery (i.e., reconfiguration). **The maximum amount of on-line testing should be incorporated as long as it does not displace normal system functions or use processing time which excessively reduces or slows the execution of normal system functions.**

Additional fault isolation and analysis can be performed in the off-line mode after completion of system recovery. Off-line testing is defined as testing of the unit functionally removed from its operational system. Off-line testing typically affords access to more information on equipment malfunctions.

An approach for arriving at the best combination of on-line and off-line testing is to use on-line passive BIT to continuously monitor the general well-being of the system and its major functions, and to use initiated BIT (in an off-line test mode) to assist in precisely locating the malfunction. Initiated BIT is also very useful in testing sections of the equipment which, if tested continuously or periodically, could disrupt normal operations.

The technical manager should be aware that testability can be incorporated into a design in two principal ways:

- Through a top-down system-level integrated approach
- Through a bottom-up building-block approach.

While it is generally agreed that the top-down system approach is highly desirable, the complexity and diversity of large systems often makes it difficult to quantify the testability evaluation criteria. Significant testability work has been done at the building block (bottom-up) or module level but attempts to extrapolate these results to higher system levels has had limited success (Ref 20). Application of sound testability design techniques and practices to a system that uses BIT will effectively reduce the BIT false alarm rate.

The current miniaturization trend and increased complexity of components has resulted in an order of magnitude increase in system functional complexity. This increased complexity adds to the importance and problems associated with implementing testability in the design. In addition, the technical manager must deal with testability impacts resulting from the use of Government Furnished Equipment (GFE) and non-standard parts in the system design.

The following paragraphs discuss a number of specific testability design concepts of interest to the technical manager of a fault tolerant system:

A. **Reduction of False Alarms & Intermittents for Redundant Circuits**

Perhaps the most obvious testability deficiency in on-line applications occurs in redundant circuits where the critical hardware is not tested unless the function fails. During a mission or operational period, one or more of the redundant circuit elements may fail without affecting prime system functions. When a transient or momentary fault occurs on one leg of a voting circuit, it is important that the other legs are operating correctly to prevent the transient from affecting the overall functional output. One technique to solve this problem is to build in independent self-check circuits for each leg of the voter to guarantee that all sections are operating properly. A check circuit for the BIT should also be considered to verify its performance. Good design practice should include provisions to log the errors which have resulted in a leg of a voting circuit being voted out. This error log would identify areas that could warrant further post-mission test and analysis.

B. **Limited Circuit Bandwidths**

Another design technique is to limit the bandwidth-limit of functional circuits in a system to those levels necessary for normal operation. Test circuitry used to monitor this function should be designed to react within the same functional bandwidth. This would eliminate false test readings caused by out of bandwidth circuit paths and would keep the test circuitry from tipping when a transient develops. This type of design has the quality of eliminating many reported "random" failures which are not really random but are attributable to this type of design deficiency. The technical manager should not regard digital functions as simple go/no-go devices; but be aware of the possibilities of using transmission line theory in digital circuits to conduct signals between points of application. New CAD/CAM/CAE type systems are available to automatically incorporate "quality" features into the design that will self-protect and frequency-tune circuit paths. This capability can also be utilized to shift the bandwidth levels or limits for electromechanical equipment to compensate for normal degradation without triggering BIT false alarms.

C. Inadequate BIT Detection Points

In many cases current systems are operating with insufficient BIT sensors; in other cases the BIT features are not being fully utilized. This condition occurs when program decisions to reduce cost cause insufficient processing capability to be allocated to BIT, or when improper initial thresholds were set for the BIT. Additional testability problems surface when the product is shipped before the testability design can be verified, with the hope that it will perform adequately in the field. The usual field result of such program actions is to increase the size of fault ambiguity groups, thereby forcing additional shop testing of gooa units (retest OKs). This leads to the obvious conclusion that the BIT false alarm problem and the testability are affected by program policy decisions as well as by the technical problems. Such decisions may have appeared expeditious at the time but were not made with a full knowledge of their impact on support tasks and system operational availability. **The technical manager must ensure that program management is aware of the technical consequences of management decisions.**

D. Operational & Environmental Data

Testability designs may be enhanced by correlating BIT failure indications with overall operational and environmental information. Operational and environmental data may consist of several factors depending on the system application. For typical airborne equipment, factors such as ambient or spot temperatures, time of day, airspeed, g-levels, primary power input voltages, turn-rate data, and cooling system parameters can be valuable in identifying the significance of BIT reports. Logic circuits can compare and correlate recorded operational and environmental data and failed BIT reports to deduce whether the failure indication represents a false alarm, an intermitt t, or a hard fault. RADC has initiated several studies which address the correlation of operational and environmental data into the BIT decision process. One f these, Smart BIT (Ref

20) has been addressed in the preceding paragraphs. RADC has also initiated a program to develop a "Time Stress Measurement Device" which would provide environmental data with an associated time tag for utilization by the BIT or post-flight maintenance activity.

E. Testability History

An important diagnostic tool for enhanced BIT is multiple reading storage. Multiple reading storage is a form of error logging. Most current BIT systems use single read-and-hold BIT fault indicators. However, if a series of readings are made and the parametric values saved which identify before fault and after fault occurrence conditions, opportunities are presented to maintenance personnel to analyze the results and discriminate between false alarms, intermittents, and hard faults.

F. Testability Data Processing

The widespread use of microprocessors and high-density VLSI memories has changed the technical base for performing testability work. It is now possible to build extensive test and diagnostic capability right into the electronic system without incurring the previous high penalties for size, weight, power, etc. This test and diagnostic capability may have to be an integral part of the functional electronics to adequately fault-detect and isolate the very complex VLSI/VHSIC circuits of the next-generation systems.

While built-in-test circuitry with expanded memories can do far more than previous BIT techniques, there may be occasions where very small, lightweight test equipment could be utilized effectively at the organizational level. The lightweight equipment may provide as much diagnostic power as previous large rack-mounted intermediate level ATE. When overall testability design integrates combined BIT and adequate [illegible] interfaces for organizational level test equipment as a "test cumulative" or integrated diagnostic system, it is possible to radically increase test and diagnostic effectiveness at a potentially much lower development cost.

Additional testability advantages can be realized when the operating system contains a data recorder. If a critical system interrupt occurs during the mission, an automatic or operator-controlled option could be provided to record the contents of pertinent system status registers so that the data could be used later to analyze the transient fault. In this manner, new insights could be gained to resolve problems previously classified as false alarms, intermittents, etc. This technological advance ultimately could be tied into a worldwide communications network which could permit expert maintenance assistance to operating system crews in remote areas.

The requirements for compatibility with external test resources, as determined by the integrated diagnostics concept, should be considered by designing proper interfaces into the system design. Usually, a combination of BIT and external test is employed for a given system. The use of external test functions can be costly and, if not properly integrated with the prime system, may result in more reliability and maintainability problems than it eliminates. However, external testing, where required and properly interfaced with the prime system design should reduce corrective maintenance time and increase system availability. Autonomous embedded test and diagnostics without the need for external test equipment should be considered wherever possible. Automatic test features can be adapted to detect (or predict) impending failures. Automatic external fault-isolation techniques, augmenting BIT, can reduce both the number of maintenance personnel and maintenance skill levels.

Another important consideration is the determination of the number and location of maintenance test points. Test points that are selected should be readily accessible for connection to external test equipment via system/equipment connectors or by special test connectors. They should also be selected with due consideration given to external test equipment implementation, and be consistent with reasonable external test equipment frequency and measurement accuracies. Test points should be "decoupled" from the external test equipment to assure that degradation of equipment performance does not occur as a result of connection to the external test equipment.

5.3.2.4 Test Methodology for Fault Tolerant Systems

This subsection discusses a number of desirable design considerations for testability. These considerations are important to establish the testability design of a fault tolerant system, and include the following:

- **Comparison Method** - An effective method for testing similar systems with similar inputs and outputs is to compare outputs and flag any gross disagreements. A maans to determine which branch is faulted and an error log entry should be mandatory.
- **Redundancy Verification** - Each redundant path should be tested individually to prevent the masking of faults in redundant items.
- **Flexing of Spares** - Periodically activate the built-in-test of the hot spares, log any errors found, and report out status before these items are needed for system operation. This will prevent a faulty unit from being switched in when the system reconfigures.
- **Voting Scheme Technique** - A typical example of a voting scheme technique is to compare output values from three different sources. Confidence is placed in that value where at least two of the three sources agree. Errors found should be logged, and the source of the erroneous value should be recorded and corrected at an appropriate maintenance interval. Since diagnostic procedures are generally designed to locate a single fault, potential exists for the occurrence of multiple faults (e.g., a stuck-at-1 in multiple locations) that can go undetected. It may be necessary to add logic or test circuitry to ensure that each state, and each state transition, occurs correctly (Ref 3).
- **Error Correction** - Detection of degraded performance in stages preceding an error-correcting function is difficult since the error-correcting function makes its preceding degraded stage appear healthy. The error-correcting functions should keep count of the number of times a correction had to be made and a record made in an error log. When a predetermined threshold count is exceeded a test signal may be injected to determine if the input stage is unacceptably degraded.
- **Multiple Redundancy** - In redundant systems which are allowed to degrade gracefully through failures of redundant elements, a test

should be established to verify that minimum acceptable system performance and redundancy levels are available at the start of a mission.

- **Echo Message** - When it is necessary to transmit long messages, the ability to echo back a message is particularly useful. This feature provides confidence that the message has been accurately received. A time out is usually set in anticipation of the echo message. If no message, or if an erroneous echo is received before the time out has elapsed, the message can be sent again and a fault flag set.
- **Redundant Bus** - Provision for a status word has been included successfully in 1553-type systems that use redundant buses. Subsystem access to the bus is completely controlled by a bus controller. Each subsystem is informed by the bus controller when to send and when to receive a message. Every time a subsystem receives such information from the bus controller, the subsystem sends a status word back to the bus controller. This status word usually contains a number of bits reflecting the health of the subsystem, the actual word-count received, the comparison results of the expected word-count, the word-count it is presently sending, etc. If the bus monitor detects an error within the bus system, it automatically switches over to the redundant bus and reports this out upon demand. Maintenance personnel can isolate a fault quickly by observing failure indications from the bus monitor as well as from the various subsystems.
- **Non-Volatile RAM** - A microprocessor's ability to access a non-volatile RAM serves a dual purpose. First, it can log fault information that may be retrieved by maintenance personnel after power has been shut off. Secondly, it can log software errors detected and trapped during on-line programming. A third possible service worth noting is the use of non-volatile RAMs to periodically check certain computed values. Power transient induced faults would then become tolerable because the processor would have to only "roll back" to the value stored at the checkpoint rather than begin the entire computation all over again.
- **Intermittent Faults** - One way to identify intermittent faults is to

log every detected occurrence into memory (preferably non-volatile memory). Once the trend of the intermittent fault is determined, effective corrective action can be taken.

- **Signal Elements** - It is often imperative that C^3I signals be sent and received in hostile jamming environments. Receivers can accurately interpret a signal even if 1/8 of its total initial format is lost. Although these receivers work extremely well, higher levels of fault detection coverage would be difficult to achieve with conventional overall wraparound tests or even quick operational checks. At close range, these systems perform perfectly without antennas or even without their power amplifiers. Elegant, localized sensitivity tests, therefore, can be built into the equipment. If the equipment is unacceptably degraded, the demodulation elements must present their own fault flag outputs.
- **Caution Indications** - Fault tolerance can be applied to a variety of system types (i.e., electrical, mechanical, hydraulic, environmental, etc). Regardless of the system type, it is customary to include a cautionary indication whenever a backup system is called into service, especially for safety-critical functions.

5.3.2.5 **Fault Detection Latency Times** - One of the most rigid demands imposed upon the testability design of fault tolerant systems is the quick response time necessary to reconfigure. Hence, the testability design process must take into account both spatial and temporal considerations for fault detection. The failure detection approach selection must be based upon the requirement for maximum acceptable failure latency. Continuous failure detection techniques should be used to monitor those functions that are mission-critical and/or affect safety and where protection must be provided against the propagation of errors through the system. Periodic testing may be used for monitoring those functions which provide backup/standby capabilities or are not mission-critical. Operator initiated testing is typically used for monitoring those functions which require operator interaction, sensor simulation, etc, or which are not easy, safe, or cost-effective to initiate automatically. The maximum permitted latency for failure detection determines the frequency at which diagnostic procedures should be run and must take into account function criticality, failure rate, possible wear-out factors, and the overall maintenance concept.

5.3.2.6 Partitioning/Levels of Fault Isolation - Fault tolerant systems may be viewed as a group of subsystems, each with varying degrees of fault tolerance (Ref 13). An ideal partition for each subsystem would result in a set of modules that contain sufficient redundancy such that the failure of one module does not degrade subsystem performance. In the event of a fault, the lower level modules can provide outputs (e.g., outputs encoded in error detection code) which can be compared by the next higher level and trigger an appropriate response (i.e., hardware reconfiguration, notification to operator, etc).

A primary function of system level testability is to distinguish between usable and non-usable resources, since this is an essential input to resource assignment. Responsibility for system level testability should be given to the same function that performs resource-assignment. When electronic circuits are properly partitioned, the subcircuit sections can be tested independently using a multiplexer or other switching process. Since the partition principle states that the number of tests required to exhaustively test the subcircuits of an electronic circuit design is fewer than the number of tests required to exhaustively test the entire circuit, partitioning will greatly reduce the testing necessary to detect and isolate faults.

For system-level testability, partitioning by function or by location is possible. An example of partitioning by function is the test of all communication links handled as one assignment, testability of all sensors handled as another, and testability of all computers handled as a third. This permits concentration of appropriate technical resources and a high degree of information-sharing when problems arise. Error recovery is usually handled as a centralized function which may require the sequencing of the recovery authority to the functional area where the errors are present. Partitioning by location means that testability of all resources in, for example, location 1 is handled within that location, testability of all resources in location 2 is handled there, etc. Obviously, location boundaries must be drawn very carefully and interfaces must be well defined. This approach avoids some administrative interface problems that usually plague the one based on function, and provides high motivation and technical expertise, in addition to the local authority for the recovery actions that are required when a unit fails. Although this latter activity may be considered a part of fault tolerance, it is in the interest of system technical managers to keep these capabilities in mind in any decision regarding top-level partitioning. Either approach can work when carried out in a dedicated manner. If location-based partitioning is used in a large system a further partition by nodes within a sector may be appropriate. In all other cases, the next partition is usually to the line replaceable element and below that to intermediate or shop level replaceable units.

Partitioning at the node level can be established in a recovery/reconfiguration mode. Recovery can take the form of simple reconfiguration at the node level, or with internal node reliability enhancement techniques such as bus ripplers, memory ripplers, master/slave/spare concepts, or reconfiguration with rollback. All three techniques can be applied with a combination of hardware and software in an integrated system. As much fault avoidance and recovery as possible should be implemented at the local level. At the higher levels, a fault tolerant operating system must provide the basic detection, isolation, and recovery mechanisms.

Fault tolerant processor applications can utilize a number of different fault detection, recovery and reconfiguration techniques. Each will result in varying levels of processing overhead and processor utilization. Processors configured as "self-checking" pairs have very effective fault detection capability. However, their inherent fault isolation capability can be poor since one processor in each pair may still be good, but the ambiguity as to which processor is good (or failed) may not easily be resolved if the quality of the self-test is suspect. Hence, the failure of one processor in a pair may effectively reduce by two the processor compliment at the system level.

Processors configured as triple modular redundant have much more effective fault isolation capability, since upon detection of a fault the node can continue to operate as a self-checking pair, with the voter used as a comparator. In addition, triple modular redundancy allows masking of transient faults, thereby reducing the incidence of false alarms. A more generalized approach which used N-modular redundancy is capable of fault detecting and isolating multiple processor failures, provided that the voter is configured as an adaptive voter (see Subsection 6.1.5).

5.3.3 Testability Checklist Questions

a. What levels of testability/diagnostics are required to meet the fault tolerance design goals for probability of success and readiness?

b. Can the BIT design (used to detect and isolate faults for performance monitoring and maintenance) be used to achieve the desired fault tolerance performance levels?

c. What additional constraints are imposed by the testability/diagnostics features of a fault tolerant design (i.e., cost, weight, complexity, volume, power, etc)?

d. Can the ratio of function circuitry failure rate to BIT circuitry failure rate be kept to 10-to-1 (as a rule of thumb) and still cover all fault tolerant system diagnostics requirements?

e. What are the time constraints for BIT performance in the operational time line?

f. Are redundant paths checked individually so as to prevent faults from being masked?

g. Are hot and cold spares periodically checked?

h. Are the test priorities for each equipment consistent with the equipment redundancy level and function criticality?

i. Has the ability to echo back a message to the sending unit been provided? Is a time out set in anticipation.of an echo message?

j. Are non-volatile RAMs used to periodically checkpoint certain computed values?

k. Does the system maintain a fault log of intermittent faults?

l. Is the system operator notified via suitable caution advisory techniques when a failure or unresolved fault degrades system performance below acceptable levels?

m. Is the size of the ambiguity group that results from FD/FI tests consistent with reconfiguration and resource sharing requirements?

n. Has an analysis been conducted to assure that system reliability has not been degraded by inadequate FD/FI in a multi-redundant circuit?

o. Has system dead-time been utilized to interleave BIT tests?

p. Has maximum fault detection latency time been considered in establishing test frequency?

q. Has the system been partitioned from a FD/FI standpoint so that faults are isolated to line replaceable units?

6 - HARDWARE & SOFTWARE FAULT TOLERANT DESIGN OPTIONS

Technical managers and designers may choose from a variety of hardware and software fault tolerant design options to satisfy C³I system reliability and availability requirements. This section provides an overview of many of these techniques and discusses their advantages, disadvantages and R/M/T impacts. It addresses the increasingly important issues of fault detection, fault avoidance, distributed processing, and levels of redundancy implementation for fault tolerance.

For many applications, reliability improvement through fault avoidance techniques often proves to be the least expensive approach to attaining a reliability goal. However, these techniques must be introduced early in the design process. They include provisions to:

- Obtain higher quality parts/components
- Increase design safety margins/parts derating
- Exercise error-reducing design practice, such as shielding and grounding
- Improve and control the operating environment through cooling, heating and isolation
- Improve user/operator proficiency.

Key elements of fault tolerance and fault avoidance are depicted in Fig. 6-1. Experience has shown that a hierarchical approach, involving the selective application of these fault tolerant design techniques, is most effective in designing fault tolerant C³I systems.

In general, fault tolerance design techniques fall into two categories: fault masking and fault reaction. In early applications, fault masking utilized multiple hardware redundancy in dual, triple or quadruple configurations. In this form, the functional interconnections remained fixed while failures consumed the components until all alternate paths were exhausted. Fault detection was not utilized in conjunction with hardware redundanc; and no intervention was made from outside the system to en-

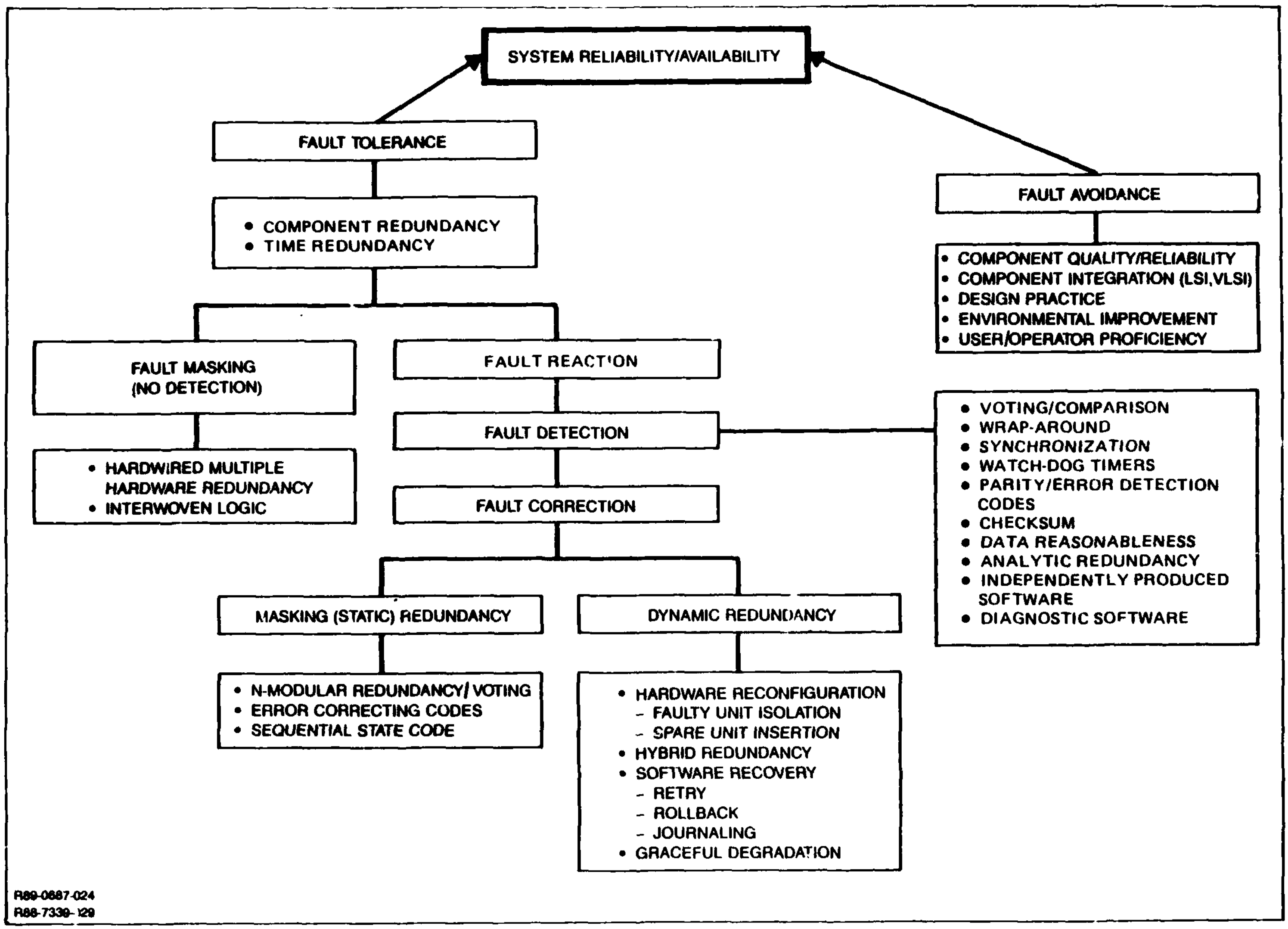


Figure 6-1. Elements of Fault Tolerance.

able switching or reconfiguration. Today, these hardware redundancy techniques are still employed, but hardware/software fault-masking often utilizes fault detection to initiate system reconfiguration. Switching to standby or spare units is an example of dynamic fault correction redundancy, whereas, the use of error detection and correction code is an example of software fault masking.

In all cases, error detection is the initial step in implementing fault reaction techniques. Once the fault is detected, the system must correct the fault or inform the operator so that an alternate means of operation may be provided. The fault correction techniques or fault reaction "strategies" can be categorized in two forms: masking redundancy or dynamic redundancy. Masking redundancy uses both detection and correction techniques and is also considered "static" in that it employs built-in hardware for detection, switching, and data error correction and requires no interaction with equipment located outside the subsystem or module. Dynamic redundancy techniques provide reconfiguration of the remaining system elements around the failed element(s). These rely on the system's ability to fault detect and isolate the failed element(s).

Specific hardware and software approaches to reconfiguration are generally tailored to meet the constraints and requirements of the particular system. The timing constraints of control systems often mandate near instantaneous reconfiguration. Other systems may allow an error to exist and to be averaged with unfailed outputs while fault isolation procedures are performed (either automatically or manually). The reconfiguration timing requirements for some C^3I applications may permit short lapses in fault coverage. For example, the loss of a radar antenna element in a phased array antenna has minimal impact on detection probability for a typical target track since even the worst case target (i.e., one with a radial velocity vector) can be detected when the sensor platform is rotated. In such cases reconfiguration need not be either instantaneous or automatic, and options might include on-line corrective maintenance, performing corrective maintenance upon completion of the mission, and in extreme cases, the deferral of the corrective maintenance to correspond with a scheduled maintenance cycle.

Typical hardware/software approaches to reconfiguration are discussed in Subsections 6.1 and 6.2. Figure 6-2 provides examples of specific reconfiguration strategies for a number of state-of-the-art C^3I systems. Subsections 6.3, 6.4, and 6.5 discuss the topics of fault avoidance, distributed architectures, and levels of implementation of fault tolerance, respectively.

Function/Application	Reconfiguration Strategy	Typical Response Time
• Digital Flight Control System		
– Processors	Hardware mid-value logic covers first failure. Error averaging followed by switchover to an analog backup for second failure.	12.5 milliseconds (1 processor cycle) for first failure. 30 milliseconds (2 processor cycles) for second failure.
– Command Sensors	Hardware mid-value logic covers first failure. Error averaging with in line failure monitor for second failure.	12.5 milliseconds for first failure. 50 milliseconds for second failure.
– Hydraulic Pump	Automatic switch-on of backup emergency power unit. Oversized accumulators provide hydraulic power while emergency unit powers up.	Approximately 2 seconds.
• Stability Augmentation System	Majority vote in analog hardware for first failure. Manual disconnect (pilot action) for second failure.	First failure – instantaneous. Second failure – several seconds.
• Environmental Control System	Manual turn off of bleed air and turn on of RAM air.	Up to 1 minute.
• Generic C^3I Platform Radar Transmitter	In-flight corrective maintenance to switch in back-up unit.	Approximately 20 minutes.
• Displays & Controls	Four operator stations supplied. Workload redistributed among remaining stations in event of a failure.	Several minutes.
• Generic Phased Array Radar	Up to 10% of transmitter/receiver antenna elements can fail before significant degradation results.	Instantaneous.

R89-0687-026

Figure 6-2. Example Reconfiguration Strategies.

6.1 Hardware Redundancy Implementation

Redundancy is the design technique of providing more than one means of accomplishing a given system function. Hardware redundancy is implemented to increase the probability of system success when the reliability of non-redundant hardware is inadequate to meet the stated qualitative and/or quantitative system requirements. The hardware added to provide an alternate means to accomplish a function need not be identical to the primary hardware, but, in general, all paths must fail before there is a system failure.

Depending on the specific application, a number of approaches are available to improve reliability through hardware redundancy. These approaches may be classified on the basis of how failures are detected and how the redundant element(s) are introduced into the system (Ref 24), and fall into one of the following broad groups:

- Active Redundancy - All redundant elements operate simultaneously
- Standby Redundancy - Alternative means of performing the function do not operate until activated on failure of the primary means of performing the function.

Techniques related to each of these classes are identified in the simplified tree-structure shown in Fig. 6-3.

Quite often, redundancy is implemented so that safety requirements can be met. These requirements are generally qualitative in nature and address the continued safe operation of a system after a failure. For example, flight control system reliability requirements including fail operational (FO), fail-operational/fail-safe (FO/FS), and fail-operational/fail-operational/fail-safe (FO^2/FS) are stated in MIL-F-9490 and imply that redundant hardware may be required. **In cases where the implementation of redundancy is being considered as a means to meet a numerical reliability requirement, it is particularly important that alternative fault avoidance techniques be examined first** (e.g., derating, design simplification, or substitution with higher quality parts) (see Subsection 6.3). Although the redundant hardware may be effective in meeting the probability of success criteria, the added hardware complexity will result

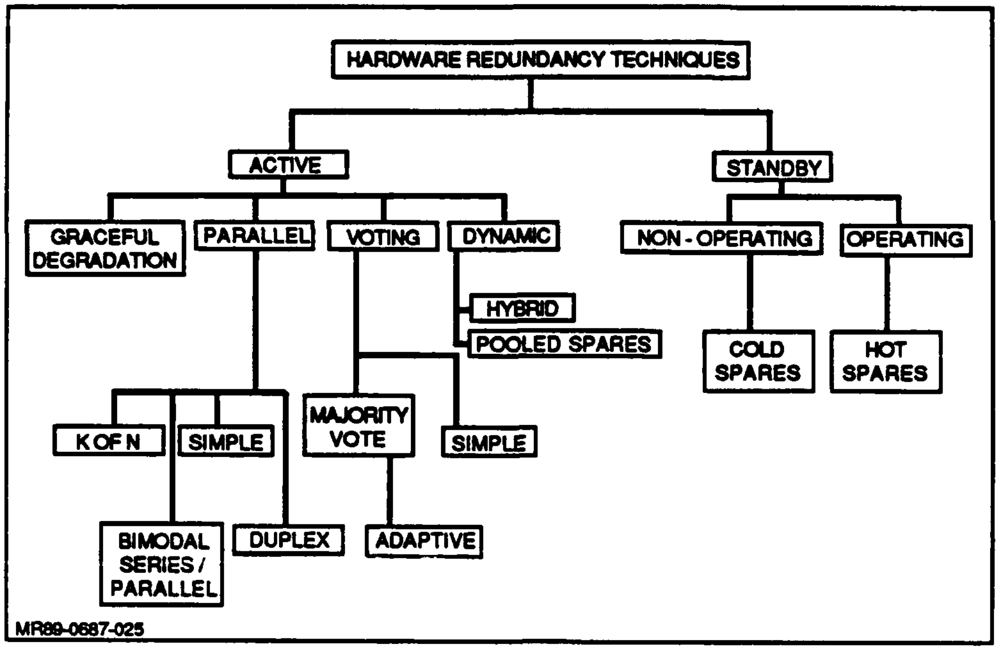
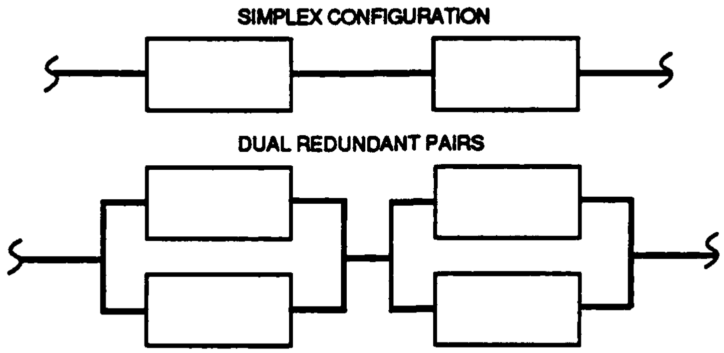


Figure 6-3. Hardware Redundancy Techniques.

in an increased serial failure rate (see Fig. 6-4). The decision to use redundant design techniques should be based on the results of a tradeoff analysis (see Section 7) involving probability of success, safety, and cost, since the additional equipment will increase maintenance costs during the operational phase of the system's life cycle.

Estimates of the inherent reliability of each functional element must be calculated early in the design process. These failure rate estimates are essential inputs to evaluate reliability math models for alternate redundancy configurations. Reliability analyses using these models are effective in reducing the number of candidate redundancy schemes capable of satisfying system reliability requirements. The mathematical models for several redundancy configurat ons are included in the subsections that follow.

Incorporating redundancy to achieve increased reliability requires an effective fault detection and isolation scheme (see Subsection 5.3). Fault isolation is necessary to prevent failures from adversely affecting other parts of a redundant network. Fault detection is used to assure the "full-up" operational status of all redundant equipment(s) at the start of



PARAMETER	CONFIGURATION	
	SIMPLEX	REDUNDANT PAIRS
PROB OF SUCCESS	0.998	0.999998
SERIAL FAILURE RATE	0.002	0.004
SERIES MTBF	500	250
MTBCF	500	5×10^5

NOTE:
- ELEMENT MTBF =1000 HOURS
- 1 HOUR MISSION TME
- ASSUMES PERFECT SWITCHING
- FAILURE PREVENTS FLOW FROM INPUT TO OUTPUT
- FAILED EQUIPMENT REPAIRED OR REPLACED BETWEEN MISSIONS

MR89-0687-027

Figure 6-4. Impact of Redundancy on MTBF.

a mission, and to inform the operator of failures that may occur during the mission. Technical managers should ensure that an FMECA is performed that is sufficiently detailed to uncover any design flaws that can result in failure propagation in redundant elements.

The penalties associated with the application of hardware redundancy includes increased maintenance, weight, volume, complexity, cost, spares, and design/development time. The increase in complexity results in the increased frequency of unscheduled maintenance and use of support resources. Thus, safety and system reliability are improved at the expense of components added to the maintenance chain. However, the increase in maintenance may be minimized by implementing reliability improvement techniques such as design simplification, component derating, and the use of more reliable components.

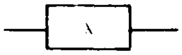
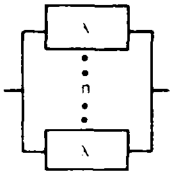
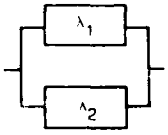
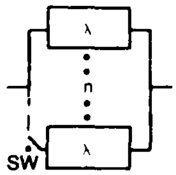
Figure 6-5 presents a summary of several hardware redundancy techniques, and includes reliability math models and equations along with associated R/M/T impacts and typical applications to current and future C^3I systems. Figure 6-6 illustrates the impact of various redundant element configurations on reliability as measured in terms of equivalent system MTBF. For the purposes of this illustration it was assumed that six elements were required for system success. Improvements in equivalent system MTBF are possible if a resource sharing (pooled spares) configuration requiring six of eight elements is implemented instead of alternate series, standby redundant, or active redundant configurations. Thus, adding a small number of spare elements operating independently of each other under software control results in a dramatic improvement in mission reliability.

The remaining paragraphs of this section deal with the technical merits and associated reliability evaluation methodology for various hardware redundancy configurations. These configurations range from active and standby redundancy to N-modular and dynamic redundancy.

6.1.1 Active Redundancy

Active (parallel) redundancy is a design technique where one (or more) continuously energized redundant element(s) is added to the basic system so that the function continues to be performed as long as one element remains operative. Simple active redundancy is configured with identical redundant elements having the same failure rate. Active redundancy configurations can include parallel redundant elements of unequal failure rates as well as series-parallel/parallel-series elements.

The reliability improvement available by use of simple active redundancy is illustrated in Fig. 6-7. In general, the system reliability gain diminishes rapidly for additional parallel redundant elements beyond triple or quadruple redundancy. As additional redundant elements are added,

FAULT TOLERANT DESIGN OPTIONS	RELIABILITY IMPACT	MAINTAINABILITY IMPACT
NON REDUNDANT (SIMPLEX)  EACH AND EVERY UNIT DEPICTED IN THE SERIES CHAIN IS REQUIRED FOR MISSION SUCCESS $R = e^{-\lambda t}$	• UNABLE TO ATTAIN HIGH SYSTEM RELIABILITY FOR SYSTEMS CONTAINING COMPLEX EQUIPMENT OR LONG DURATION OPERATIONS • ACCEPTABLE SYSTEMS RELIABILITY MAY BE ACHIEVED WITH HIGH RELIABILITY EQUIPMENT & SHORT OPERATING TIMES	• MINIMAL SPARES & MAINTENANCE PERSONNEL REQUIRED COMPARED WITH REDUNDANT SYSTEMS
ACTIVE (SIMILAR) REDUNDANCY  CONSISTS OF A NUMBER(n) OF IDENTICAL, CONTINUOUSLY OPERATING UNITS & ONLY ONE IS REQUIRED FOR MISSION SUCCESS $R = 1 - (1 - e^{-\lambda t})^n$	• HIGH SYSTEMS RELIABILITY CAN BE ATTAINED WITHOUT SYSTEMS INTERRUPTION • POTENTIAL COMMON FAILURE MODE (OR THREAT) CAN IMPACT ALL REDUNDANT UNITS	• SEVERE IMPACT ON SPARES & MAINTENANCE PERSONNEL SINCE ALL UNITS ARE OPERATING CONTINUOUSLY
ACTIVE (DISSIMILAR) REDUNDANCY  CONTINUOUSLY OPERATING UNITS HAVE UNEQUAL FAILURE RATES (λ) & ONLY ONE IS REQUIRED FOR MISSION SUCCESS (SAME AS ACTIVE (SIMILAR) BUT NON-IDENTICAL UNITS UTILIZED) $R = 1 - (1 - e^{-\lambda_1 t})(1 - e^{-\lambda_2 t})$	• HIGH SYSTEMS RELIABILITY CAN BE ATTAINED WITHOUT SYSTEMS INTERRUPTION • NORMALLY LESS SUSCEPTIBLE TO COMMON FAILURE MODE OR THREAT ENVIRONMENT	• COMPLICATION OF SPARING & MAINTENANCE OF DIFFERENT UNITS
STANDBY (SIMILAR) REDUNDANCY  CONSISTS OF A SINGLE CONTINUOUSLY OPERATING PRIMARY UNIT, A NUMBER (n) QUIESCIENT IDENTICAL UNIT(s) AND A SWITCH. THE QUIESCIENT/STANDBY UNIT(S) ARE NOT OPERATIONAL UNTIL SWITCHED IN UPON FAILURE OF THE PRIMARY UNIT. ONLY ONE UNIT IS REQUIRED FOR MISSION SUCCESS $R = e^{-\lambda t}\left[1 + \lambda t + \frac{(\lambda t)^2}{2!} \cdots + \frac{(\lambda t)^n}{n!}\right] R_{SW}$ R87-3537-014(1/2)(T)	• VERY HIGH SYSTEMS RELIABILITY CAN BE ACHIEVED COMPARED TO ACTIVE REDUNDANCY IF SYSTEMS INTERRUPT FOR STANDBY UNIT "WARM-UP" & "SWITCH-IN" IS ACCEPTABLE • POTENTIAL COMMON FAILURE MODE OR THREAT CAN IMPACT ALL REDUNDANT UNITS	• MINIMAL SPARES & MAINTENANCE PERSONNEL REQUIRED FOR HIGH RELIABLE SYSTEMS SINCE STANDBY UNITS ARE NON OPERATIVE & ARE LESS LIKELY TO FAIL

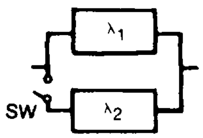
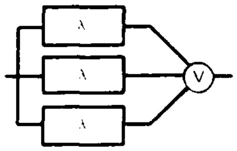
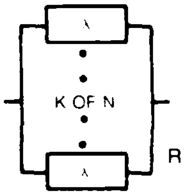
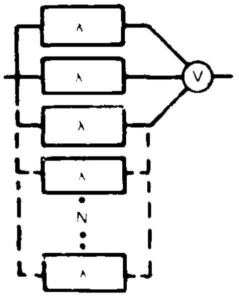
NOTE: R = RELIABILITY; λ = FAILURE RATE, t = OPERATING TIME (HOURS)

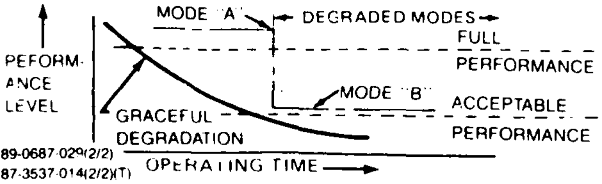
R89-0687-029(1/2)
R88-7339-033(1/2)

	MAINTAINABILITY IMPACT	TESTABILITY IMPACT	TYPICAL APPLICATIONS
ILITY TIONS CE MENT	• MINIMAL SPARES & MAINTENANCE PERSONNEL REQUIRED COMPARED WITH REDUNDANT SYSTEMS	• SELF CHECK CAPABILITY SHOULD BE PROVIDED ON A NON-SYSTEMS INTERRUPT BASIS • LESS COMPLEX FAULT DETECTION/ISOLATION COMPARED TO REDUNDANT SYSTEMS	• LOW CRITICALITY APPLICATIONS OR WHERE REPAIR CAN BE RAPIDLY ACCOMPLISHED TO MINIMIZE DOWNTIME • RELIABLE EQUIPMENT WITH SHORT OPERATING TIME • SYSTEMS WITH CONSTRAINTS IN COST, WEIGHT, VOLUME
NED R UNITS	• SEVERE IMPACT ON SPARES & MAINTENANCE PERSONNEL SINCE ALL UNITS ARE OPERATING CONTINUOUSLY	• DIFFICULT TO DETECT A FAULT IN REDUNDANT ELEMENTS WITHOUT A REDUNDANCY MANAGEMENT SCHEME SUCH AS COMPARISON MONITORING, VOTING, ETC • SELF-TEST CAPABILITY SHOULD BE PROVIDED FOR EACH REDUNDANT ELEMENT	• HIGH CRITICALITY APPLICATIONS WHERE REPAIR CANNOT BE ACCOMPLISHED AND WHERE SYSTEMS OPERATION CANNOT BE INTERRUPTED • COMPUTER PROCESSING, COMMUNICATIONS NETWORKS
INED ION T	• COMPLICATION OF SPARING & MAINTENANCE OF DIFFERENT UNITS	• DIFFICULT TO DETECT A FAULT IN REDUNDANT ELEMENTS WITHOUT A REDUNDANCY MANAGEMENT SCHEME SUCH AS COMPARISON MONITORING, VOTING, ETC. • SELF-TEST CAPABILITY SHOULD BE PROVIDED FOR EACH REDUNDANT ELEMENT • ADDITIONAL SOFTWARE TESTING REQUIRED	• HIGH CRITICALITY APPLICATIONS WHERE REPAIR CANNOT BE ACCOMPLISHED OR WHERE SYSTEMS OPERATION CANNOT BE INTERRUPTED • APPLICATIONS WHERE CONCERNS EXIST FOR COMMON MODE FAILURE OR THREAT ENVIRONMENT
BE OR N NITS	• MINIMAL SPARES & MAINTENANCE PERSONNEL REQUIRED FOR HIGH RELIABLE SYSTEMS SINCE STANDBY UNITS ARE NON OPERATIVE & ARE LESS LIKELY TO FAIL	• DIFFICULT TO DETECT A FAULT IN REDUNDANT ELEMENTS WITHOUT A REDUNDANCY MANAGEMENT SCHEME SUCH AS COMPARISON MONITORING VOTING, ETC • SELF-TEST CAPABILITY SHOULD BE PROVIDED FOR EACH REDUNDANT ELEMENT	• HIGH CRITICALITY APPLICATIONS WHERE REPAIR CANNOT BE ACCOMPLISHED & WHERE SYSTEMS INTERRUPT FOR "SWITCH-IN" IS ACCEPTABLE

Figure 6-5. Hardware Redundancy Techniques (Sheet 1 of 2).

2

FAULT TOLERANT DESIGN OPTIONS	RELIABILITY IMPACT	MAINTAINABILITY IMPACT
STANDBY (DISSIMILAR) REDUNDANCY  THE PRIMARY AND STANDBY UNITS ARE DISSIMILAR, HAVING UNEQUAL FAILURE RATES (λ). ONLY ONE UNIT IS REQUIRED FOR MISSION SUCCESS. $R = \left[e^{-\lambda_1 t} + \frac{\lambda_1}{\lambda_2 - \lambda_1}\right]\left[e^{-\lambda_1 t} - e^{-\lambda_2 t}\right] R_{SW}$	• HIGHER SYSTEMS RELIABILITY CAN BE ACHIEVED COMPARED TO ACTIVE REDUNDANCY WITH SYSTEMS INTERRUPT FOR STANDBY UNIT "WARM-UP" & "SWITCH-IN" • NORMALLY LESS SUSCEPTIBLE TO COMMON FAILURE MODE OR THREAT ENVIRONMENT	• COMPLICATION OF SPARING & MAINTENANCE OF DIFFERENT UNITS COMPARED WITH SIMILAR STANDBY REDUNDANCY
VOTING REDUNDANCY  ELEMENTS OUTPUT STATE IS DETERMINED BY STATE OF MAJORITY OF INPUTS DETERMINED BY VOTER (V) $R_{2\text{ of }3} = R_V\left[e^{-3\lambda t} + 3e^{-2\lambda t}(1-e^{-\lambda t})\right]$	• CAN PROVIDE A SIGNIFICANT GAIN IN SYSTEM RELIABILITY FOR SHORT MISSION DURATIONS • POTENTIAL COMMON FAILURE MODE OR THREAT CAN IMPACT ALL REDUNDANT ELEMENTS • REQUIRES VOTER RELIABILITY SIGNIFICANTLY BETTER THAN ELEMENT RELIABILITY • SYSTEM OPERATION CONTINUES UNINTERRUPTED DUE TO VOTING LOGIC PROVIDING A HIGH CONFIDENCE OF MASKING A SINGLE FAULTY ELEMENT	• SEVERE IMPACT ON SPARES & MAINTENANCE PERSONNEL SINCE ALL UNITS ARE OPERATING CONTINUOUSLY
K OF N CONFIGURATIONS  OF N IDENTICAL ACTIVE UNITS, K UNITS MUST FUNCTION FOR MISSION SUCCESS (e.g., 2 of 3, 3 of 4, etc.) $R = \sum_{i=k}^{n} \binom{n}{i}\left(e^{-\lambda t}\right)^{i}\left(1 - e^{-\lambda t}\right)^{n-i}$	• HIGH RELIABILITY CAN BE ACHEIVED WITHOUT SYSTEMS INTERRUPTION AND WITH MODEST INCREASE IN SYSTEM RESOURCES	• SEVERE IMPACT ON SPARES & MAINTENANCE PERSONNEL SINCE ALL UNITS ARE CONTINUOUSLY OPERATING
HYBRID REDUNDANCY  DEFECTIVE ACTIVE UNIT(S) DETECTED BY VOTER (V) AND REPLACED BY (N) STANDBY SPARE UNIT(S)	• VERY HIGH RELIABILITY CAN BE ACHIEVED WITHOUT SYSTEMS INTERRUPTION FOR VERY LONG MISSION DURATIONS • PROVIDES HIGH CONFIDENCE IN THE CONTINUED ABILITY TO MASK FAULTS BY REPLACING FAULTY "VOTED OUT" UNITS	• SEVERE IMPACT ON SPARES & MAINTENANCE PERSONNEL DUE TO MULTIPLE ACTIVE UNITS • IDEAL CONFIGURATION FOR A DEFERRED MAINTENANCE POLICY

ACCEPTABLE DEGRADED MODES OF OPERATION & GRACEFUL DEGRADATION

R89-0687-029(2/2)
R87-3537-014(2/2)(T)
R88-7339-033(2/2)
	• NORMALLY, SYSTEMS WITH DEGRADED MODES OR GRACEFUL DEGRADATION CAN ACHIEVE HIGH RELIABILITY LEVELS WITH MINIMAL INCREASES IN HARDWARE RESOURCES	• MINIMAL SPARES & MAINTENANCE PERSONNEL REQUIRED FOR HIGH RELIABLE SYSTEMS COMPARED WITH OTHER REDUNDANCY TECHNIQUES
		• IDEAL CONFIGURATION FOR A DEFERRED MAINTENANCE POLICY

	MAINTAINABILITY IMPACT	TESTABILITY IMPACT	TYPICAL APPLICATIONS
OR	• COMPLICATION OF SPARING & MAINTENANCE OF DIFFERENT UNITS COMPARED WITH SIMILAR STANDBY REDUNDANCY	• SELF-TEST CAPABILITY SHOULD BE PROVIDED FOR EACH REDUNDANT ELEMENT	• HIGH CRITICALITY APPLICATIONS WHERE REPAIR CANNOT BE ACCOMPLISHED & WHERE SYSTEMS INTERRUPT FOR "SWITCH-IN" IS ACCEPTABLE • APPLICATIONS WHERE CONCERNS EXIST FOR COMMON MODE FAILURES OR THREAT ENVIRONMENT
LITY G E	• SEVERE IMPACT ON SPARES & MAINTENANCE PERSONNEL SINCE ALL UNITS ARE OPERATING CONTINUOUSLY	• SELF-TEST CAPABILITY SHOULD BE PROVIDED FOR EACH REDUNDANT ELEMENT	• HIGH CRITICALITY APPLICATIONS WHERE REPAIR CONNOT BE ACCOMPLISHED AND WHERE SYSTEMS OPERATION CANNOT BE INTERRUPED
	• SEVERE IMPACT ON SPARES & MAINTENANCE PERSONNEL SINCE ALL UNITS ARE CONTINUOUSLY OPERATING	• DIFFICULT TO DETECT A FAULT IN REDUNDANT ELEMENTS WITHOUT A REDUNDANCY MANAGEMENT SCHEME SUCH AS COMPARISON MONITORING ETC • SELF-TEST CAPABILITY SHOULD BE PROVIDED FOR EACH REDUNDANT ELEMENT	• HIGH CRITICALITY APPLICATIONS WHERE REPAIR CANNOT BE ACCOMPLISHED AND WHERE SYSTEM OPERATION CANNOT BE INTERRUPTED
ED ERY	• SEVERE IMPACT ON SPARES & MAINTENANCE PERSONNEL DUE TO MULTIPLE ACTIVE UNITS • IDEAL CONFIGURATION FOR A DEFERRED MAINTENANCE POLICY	• DIFFICULT TO DETECT A LATENT FAULT IN REDUNDANT ELEMENTS WITHOUT A REDUNDANCY MANAGEMENT SCHEME • SELF TEST CAPABILITY SHOULD BE PROVIDED FOR EACH REDUNDANT ELEMENT	• HIGH CRITICALITY APPLICATIONS NORMALLY OF LONG MISSION DURATION WHERE HIGH CONFIDENCE IN THE ABILITY TO MASK FAULTY OUTPUTS IS ESSENTIAL
DES E	• MINIMAL SPARES & MAINTENANCE PERSONNEL REQUIRED FOR HIGH RELIABLE SYSTEMS COMPARED WITH OTHER REDUNDANCY TECHNIQUES • IDEAL CONFIGURATION FOR A DEFERRED MAINTENANCE POLICY	• SYSTEM SHOULD BE DESIGNED TO DETECT A THRESHOLD ABOVE THE MINIMUM ACCPETABLE PERFORMANCE LEVEL	• RESTRICTED TO THOSE TECHNICAL AREAS WHERE THIS APPROACH IS APPLICABLE I E PHASED ARRAY RADARS SOLAR ARRAYS I R SENSORS ETC

Figure 6-5. **Hardware Redundancy Techniques (Sheet 2 of 2).**

2

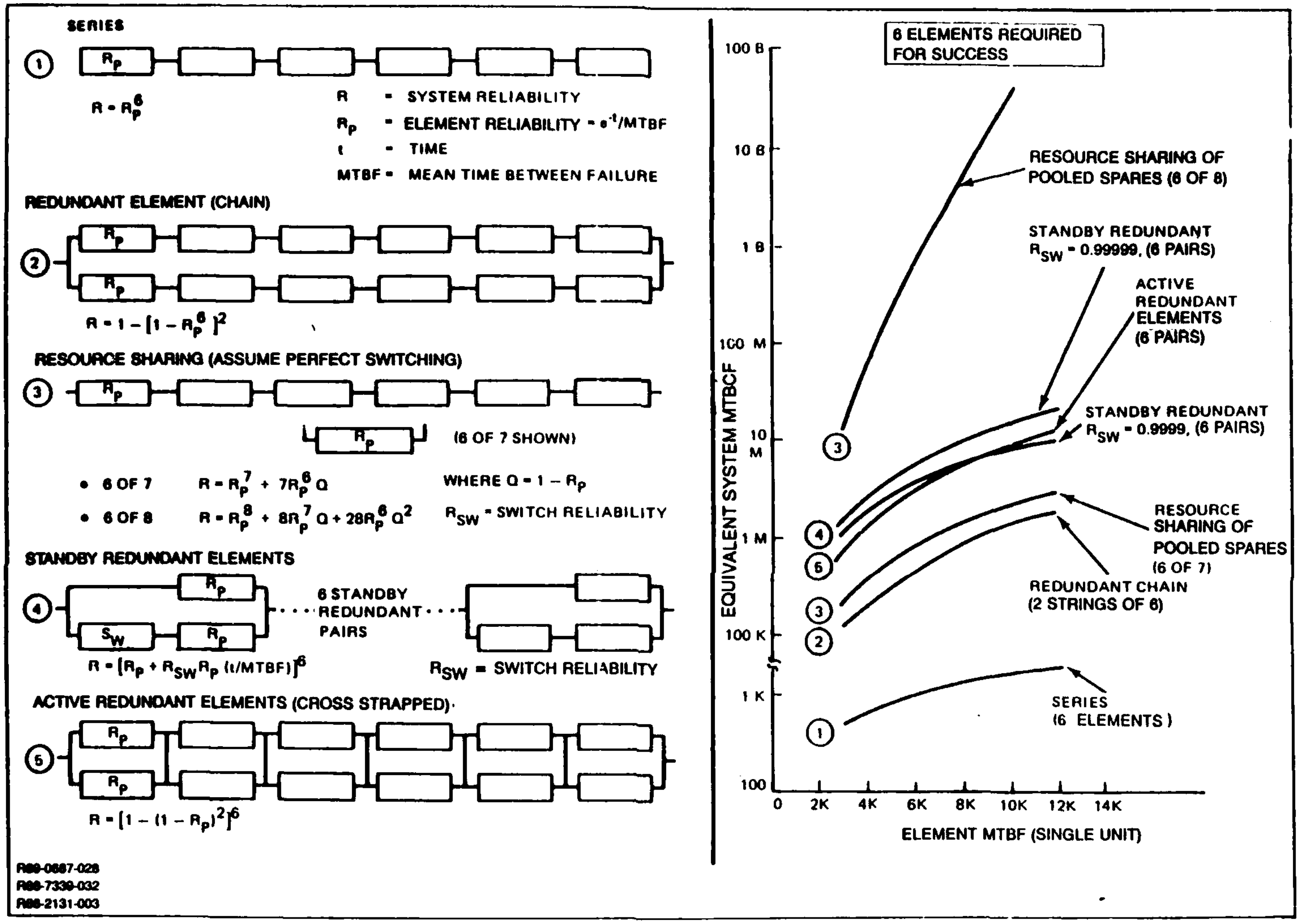
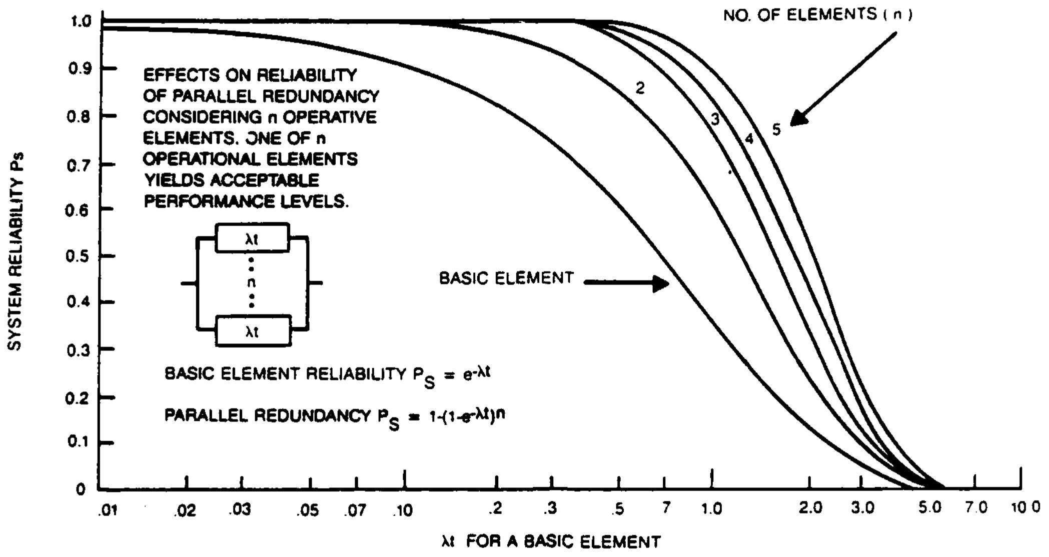


Figure 6-6. Equivalent MTBCF of Various Redundancy Configurations.



EFFECTS ON RELIABILITY OF PARALLEL REDUNDANCY CONSIDERING n OPERATIVE ELEMENTS. ƆNE OF n OPERATIONAL ELEMENTS YIELDS ACCEPTABLE PERFORMANCE LEVELS.

BASIC ELEMENT RELIABILITY $P_S = e^{-\lambda t}$

PARALLEL REDUNDANCY $P_S = 1\text{-}(1\text{-}e^{-\lambda t})^n$

ACTIVE REDUNDANCY

ADVANTAGES

- SIMPLICITY
- SIGNIFICANT GAIN IN RELIABILITY FROM NONREDUNDANT ELEMENT
- APPLICABLE TO BOTH ANALOG AND DIGITAL CIRCUITRY

DISADVANTAGES

- INCREASE IN UNSCHEDULED MAINTENANCE FREQUENCY
- DIFFICULT TO PREVENT FAILURE PROPAGATION
- MAY PRESENT CIRCUIT DESIGN PROBLEMS
- WEIGHT, COST, COMPLEXITY
- LOAD SHARING MUST BE CONSIDERED

R86-√687-030
R86-7339-034
R86-2131-004

Figure 6-7. System Reliability for Simple Active Redundancy Configurations.

the incremental increase in system MTBF diminishes. For the simple parallel case, the greatest gain (equivalent to a 50% increase in the system MTBF) is achieved by adding the first redundant element.

Designers must exercise caution in selecting the redundancy techniques to be used for a specific application. For example, consider the parallel-series active redundant configuration shown in Fig. 6-8. A parallel-series element arrangement of this type is commonly used in cases where a configuration is designed to be tolerant of opposing failure modes (e.g., fail-open/fail-short). The reliability gain for the configuration of four identical elements is compared with that of a single element configuration. This figure illustrates that high-reliability gain can be achieved when parallel-series redundancy is selected at a normalized time (t/MTBF)

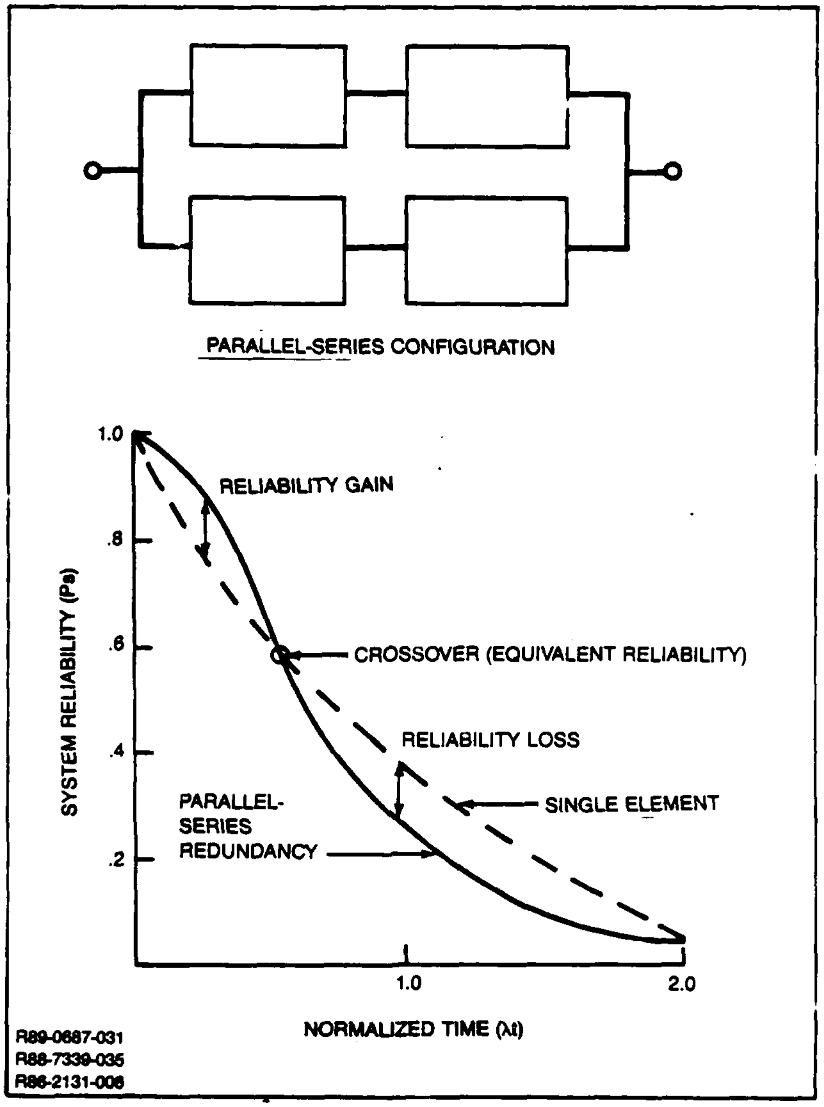


Figure 6-8. Parallel-Series Redundancy Reliability Gain.

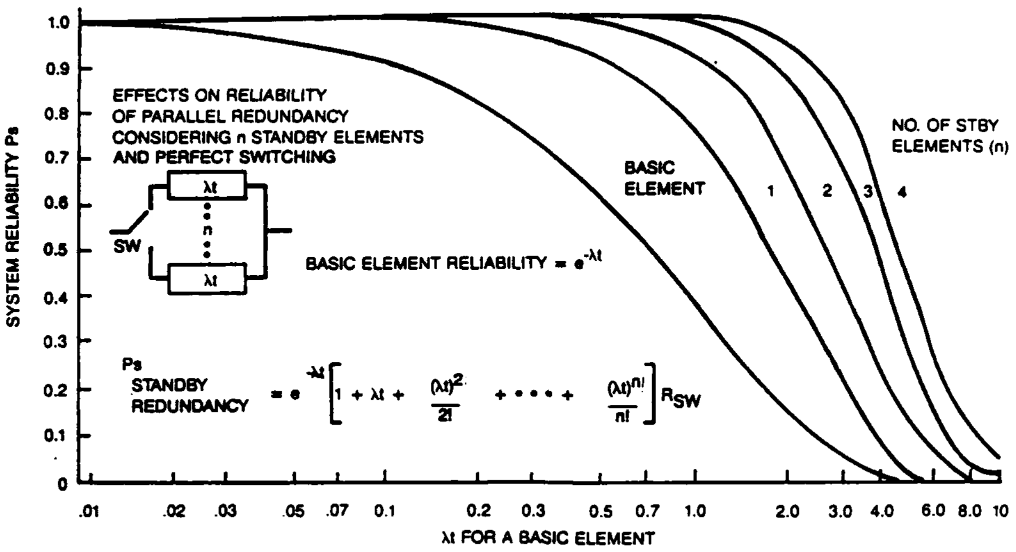
of less than 0.5. Comparing the two configurations indicates that there is a significant reliability decrement over the single element configuration when operating at a normalized time greater than 0.5 (Ref 1). This situation, where the parallel-series redundant reliability function crosses below the reliability function of the single element configuration, indicates that under certain conditions it might be advantageous to consider a series or single element configuration or an alternate reliability scheme. **Technical managers should insure that accurate reliability models are developed and evaluated so that alternate hardware architectures and redundancy schemes may be compared and traded.**

6.1.2 Standby Redundancy

Standby redundancy is a design technique where an alternate redundant means of performing the function is switched in when it is determined that a failure has occurred in the primary element(s). This differs from active redundancy in that the redundant unit or element is not operating until switched into the system as a substitute for the failed primary unit. Switching mechanisms, therefore, are always required to activate standby redundant units and disengage failed primary elements. The switch is either under the control of a subsystem that monitors the status of the redundant equipment, or the switch itself performs the monitoring function. In either case the monitoring device and/or switch can fail. If the monitoring device fails, subsequent failure of an operational unit will not be detected and system failure may result. In addition, the monitoring device may trigger a false alarm and cause the system to reconfigure, when in fact no failure has occurred. Failures of the switching devices must be considered, because the device can either fail to switch when required, or fail in a way that results in a false switch (Ref 2). Technical managers should make certain that the failure modes and effects of switches and monitoring devices are carefully considered in cases where standby or active redundancy is to be used in fault tolerant systems.

From a maintenance viewpoint, standby redundancy is attractive because the standby elements are less susceptible to failure, since they are not operating until switched in. As a result, standby elements will exhibit failure rates that reflect a reduced duty cycle when compared to that of primary units. Therefore, higher system reliability can be achieved with standby redundancy if system complexity and system interrupt due to warm-up and switching time penalties are acceptable. Although only one redundant element may be required to operate in the system, the system must contain self-test capability for all elements to assure fault detection capability.

The potential system reliability improvement (excluding reliability of switching elements) through simple standby redundancy is illustrated in Fig. 6-9. The curves relate system reliability (probability of mission



STANDBY REDUNDANCY

ADVANTAGES

- SIGNIFICANT GAIN IN RELIABILITY OVER NONREDUNDANT SYSTEM
- APPLICABLE TO BOTH ANALOG AND DIGITAL CIRCUITRY
- STANDBY UNITS ARE LESS PRONE TO FAILURE THAN ACTIVE REDUNDANT UNITS

DISADVANTAGES

- INCREASED COMPLEXITY DUE TO SENSING AND SWITCHING
- RELIABILITY GAINS ARE LIMITED BY . FAILURE MODES OF SENSING AND SWITCHING DEVICES
- DELAY DUE TO SENSING AND SWITCHING
- INCREASE IN UNSCHEDULED MAINTENANCE FREQUENCY

R89-0687-032
R86-2131-007

Figure 6-9. System Reliability for Simple Standby Redundancy.

success) to the reliability of individual standby redundant parallel elements as a function of the basic element failure rate (λ) multiplied by the mission time (t). The plot indicates that the system reliability gain for additional standby redundant elements decreases rapidly for additions beyond a few parallel elements. The required number of standby elements (n) can be determined by entering the abscissa of the chart at a point equal to the time period of interest multiplied by the basic element failure rate, and proceeding to the allocated reliability requirement.

Figure 6-9 also includes the mathematical models for system reliability and system MTBF. Note that the system MTBF increases in direct proportion to the number of standby elements added. Thus, by adding more standby elements the system MTBF can be significantly increased over that of a basic series element.

Given the same active element failure rate and the same number of redundant elements, standby redundancy generally provides a system probability of success and system MTBF that is greater than that for active redundancy. However, standby redundancy does add switching complexity and reconfiguration (time) penalties. Figure 6-10 is a plot of the system reliability comparing a two-element configuration of simple active

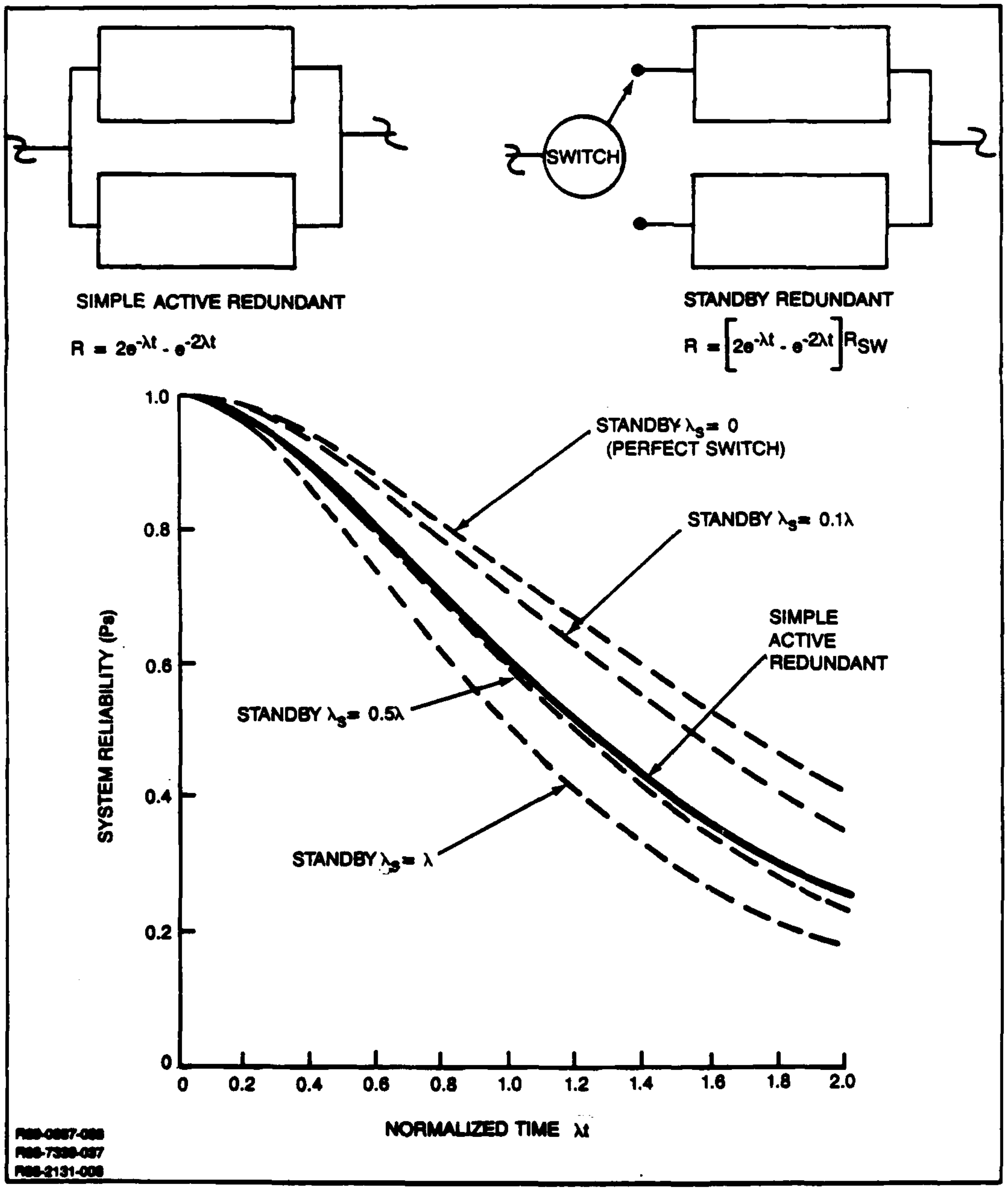


Figure 6-10. Simple Active Redundancy vs Standby Redundancy with Imperfect Switching.

redundancy (see Subsection 6.1) with that of standby redundancy having a perfect switch (zero failure-rate) and with the switch failure rates (λ) ranging from 10% to 100 % of the failure rate of the active element. This illustration shows that standby redundancy provides improved system reliability over active redundancy when the switch failure rate is low, and decreased reliability when the switch failure rate is greater than 50% of an active element.

The application of standby redundancy is not without penalties. It increases weight, volume, complexity, cost, and impacts development schedules. In addition to maintenance cost increases for repair of the additional elements, for certain unique applications the reliability of the standby redundant configuration may actually be less than that of a single element. This is due to the unreliability of switching or other peripheral devices needed to implement the standby redundant element. Care must b^ exercised to ensure that potential reliability gains are not offset by increased failure rates due to switching devices, error detectors, and other peripheral devices needed to implement the standby redundancy configurations.

Standby redundancy is attractive in those applications where repair of failed units can be accomplished while system operation continues. Based on a continuous or comparative monitoring signal or indication, a failed unit can be either automatically or manually switched over to the standby unit. While system operation continues, the failed unit can be replaced or BIT used to isolate the failed module or piece part. Groundbased and large C^3I airborne weapons systems, such as AWACS and Joint STARS, are examples of systems that utilize on-line repair techniques to enhance availability.

6.1.3 Voting Redundancy

Voting redundancy is a design technique in which the element's output state is determined by a voter or comparator that compares or analyzes the state of the majority of the outputs. Generalized approaches to voting redundancy are illustrated in Fig. 6-11. In voting redundancy, faults are statically masked because the agreeing outputs are selected by

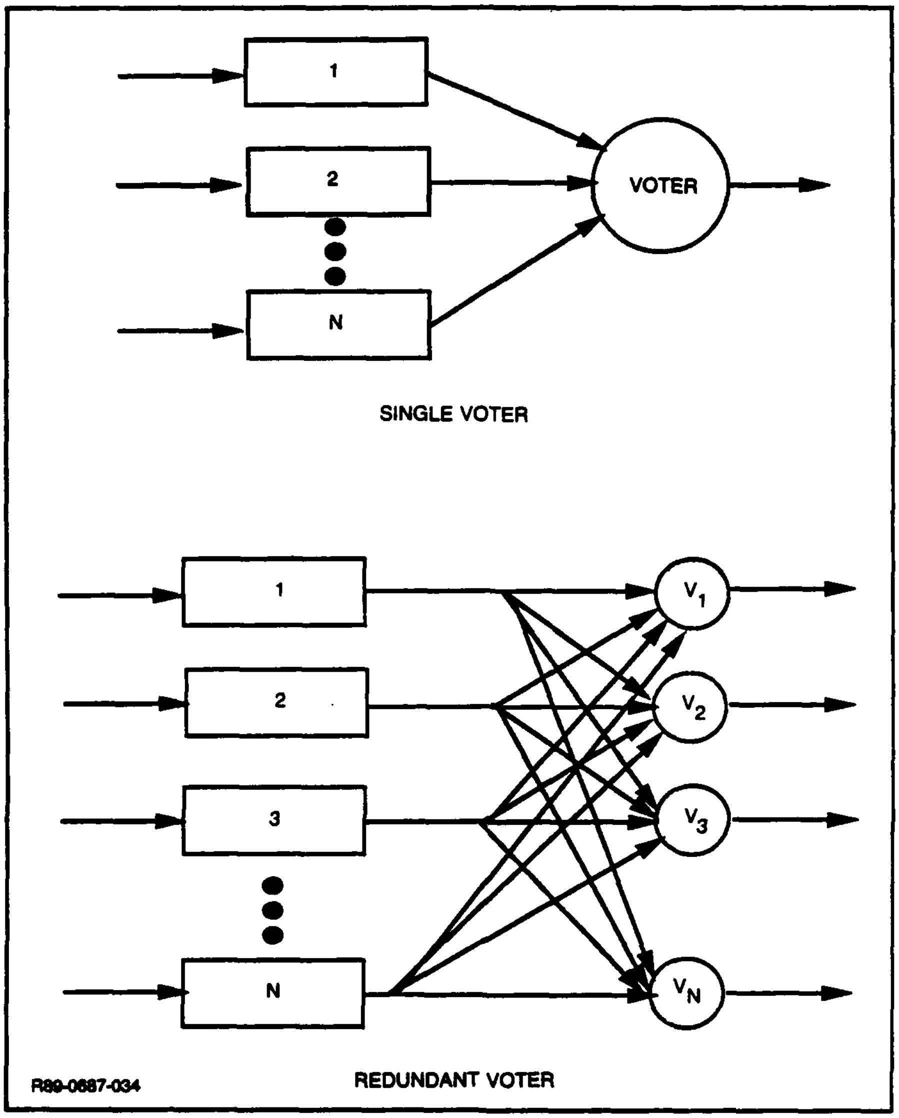


Figure 6-11. Generalized Approach to Voting Redundancy.

the voter and the faulty outputs are ignored. Thus, most agreeing outputs (presumed to be good) allow continuation of the system function without interruption. Voting redundancy must be configured with an odd number of elements to avoid the possibility of an uncertain state resulting from a tie-vote ambiguity. Minimum element implementation, called triple modular redundancy (TMR), outputs the result of two or more of three agreeing outputs by its voter (see Fig. 6-12). In TMR, the second failure results in system failure inasmuch as the remaining good module may be outvoted by the failed modules. A more general implementation,

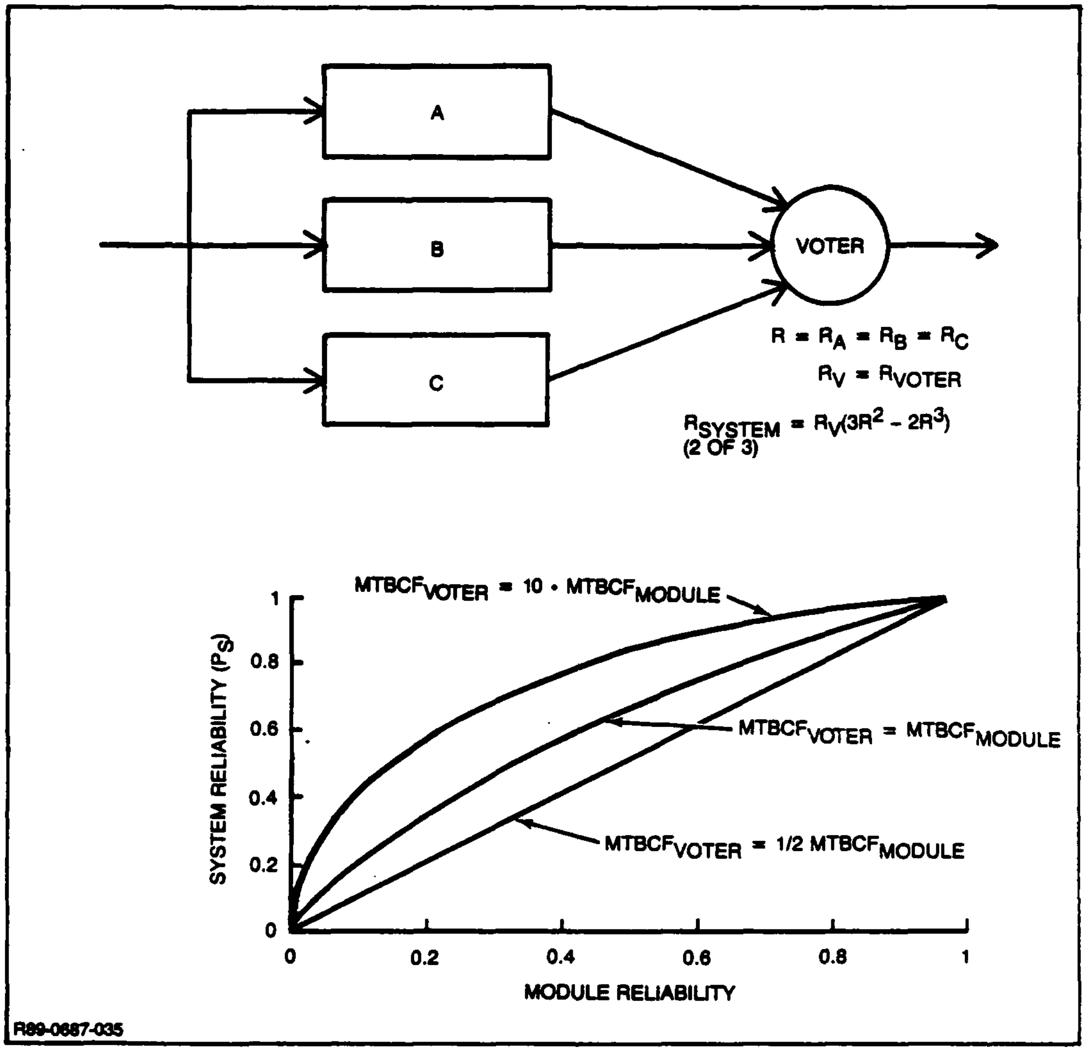

A
B
C
VOTER
$R = R_A = R_B = R_C$
$R_V = R_{VOTER}$
$R_{SYSTEM\,(2\,OF\,3)} = R_V(3R^2 - 2R^3)$
$MTBCF_{VOTER} = 10 \cdot MTBCF_{MODULE}$
$MTBCF_{VOTER} = MTBCF_{MODULE}$
$MTBCF_{VOTER} = 1/2\ MTBCF_{MODULE}$
SYSTEM RELIABILITY (P_S)
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
MODULE RELIABILITY
R89-0687-035

Figure 6-12. System Reliability for Majority Voting Redundancy.

N-modular redundancy (NMR), outputs the majority of N element outputs that agree. Voting may be applied to analog and digital signals and is commonly applied at the module level.

Voting on analog signals is almost always performed in the analog domain since the use of multiple analog-to-digital converters followed by bit-by-bit comparisons is not adequate. After analog-to-digital conversation, the least significant bits often do not agree; hence, voting in the digital domain may lead to false alarms even when all devices are functioning nominally. Analog voting techniques include the use of the median of an odd number of analog values (Ref 3), or the mean of the two most similar signals.

To prevent timing problems and false outputs, the synchronization of the voter with signals from the redundant devices is important. The synchronizaticn may be accomplished by using a common clock; however, the clock must be fault tolerant to prevent single point failure. Another technique involves the use of a synchronizing voter (Ref 34).

The penalty associated with N-modular redundancy includes the complexity (N) times the basic hardware complexity (cost, weight, volume, and power), plus the complexity of the voter. The voter may also cause a signal propagation delay, and additional performance overhead often results from the need to synchronize the arrival of the inputs at the voter. To achieve the reliability potential of NMR configurations it is important to prevent the voter from becoming a single point failure. This can be overcome by introducing fault avoidance and fault tolerance techniques into the voter design (see Subsection 6.3).

Complex systems can be designed such that individual subsystems form NMR configurations. Systems composed of a series of NMR groups can withstand more failures than a configuration made up of large replicated modules. Caution must be exercised so that subgroups are not formed at arbitrarily small levels, because the added complexity and part count of the voter mechanisms might negate the reliability gain.

In the Software Implemented Fault Tolerance (SIFT) system, the local executive detects an error when it obtains different output values for the same task iteration from different processors (Ref 13). The local executive reports all such errors to an error reporting task that performs a preliminary analysis of these errors and reports status to the global executive. If the global executive determines that a component has failed it signals the local reconfiguration task at the local executive level, and the local executive controls the reconfiguration of its resources.

To prevent the corruption of data, a technique that uses checkpoint software may be used. With checkpoint software the primary data space is copied to the secondary whenever the data space is to change. In the event of a failure, the backup process can automatically recreate a data space identical to the primary data space at the time of failure (Ref 13).

6.1.4 K of N Configurations

A K out of N configuration consists of a total of N elements, of which at least K elements must be operating properly for the system to function. All N elements in the configuration are generally operating in parallel, similar to the operation of a system configured in active parallel redundancy. However, instead of requiring only one of the N elements to function (as in active parallel redundancy), this configuration requires at least K elements to be functioning for system success. A K out of N configuration is, in essence, a voting configuration with perfect switching and voting. Figure 6-13 depicts the system probability of success for typical examples of K out of N redundancy.

Examples of K out of N configurations include a spacecraft designed such that attitude control can be maintained with any eight (or more) out of sixteen thrusters functioning. For aircraft platform stabilization, an integrated inertial reference assembly can be designed such that any three or more of six gyros and any two or more of four accelerometers provide accurate inertial reference data.

6.1.5 Dynamic Redundancy

A powerful and increasingly popular approach to increase system reliability involves implementing redundant elements in such a way that they may be rearranged (either automatically or manually) to provide continued operation of a function. This technique is referred to as "dynamic redundancy" and deals with the reconfiguration of system elements in response to failures detected either by devices internal to the failed unit, or by detection of erroneous output from the failed element (Ref 3).

Successful implementation of dynamic redundancy depends heavily upon the fault detection and fault isolation capability in the design. The

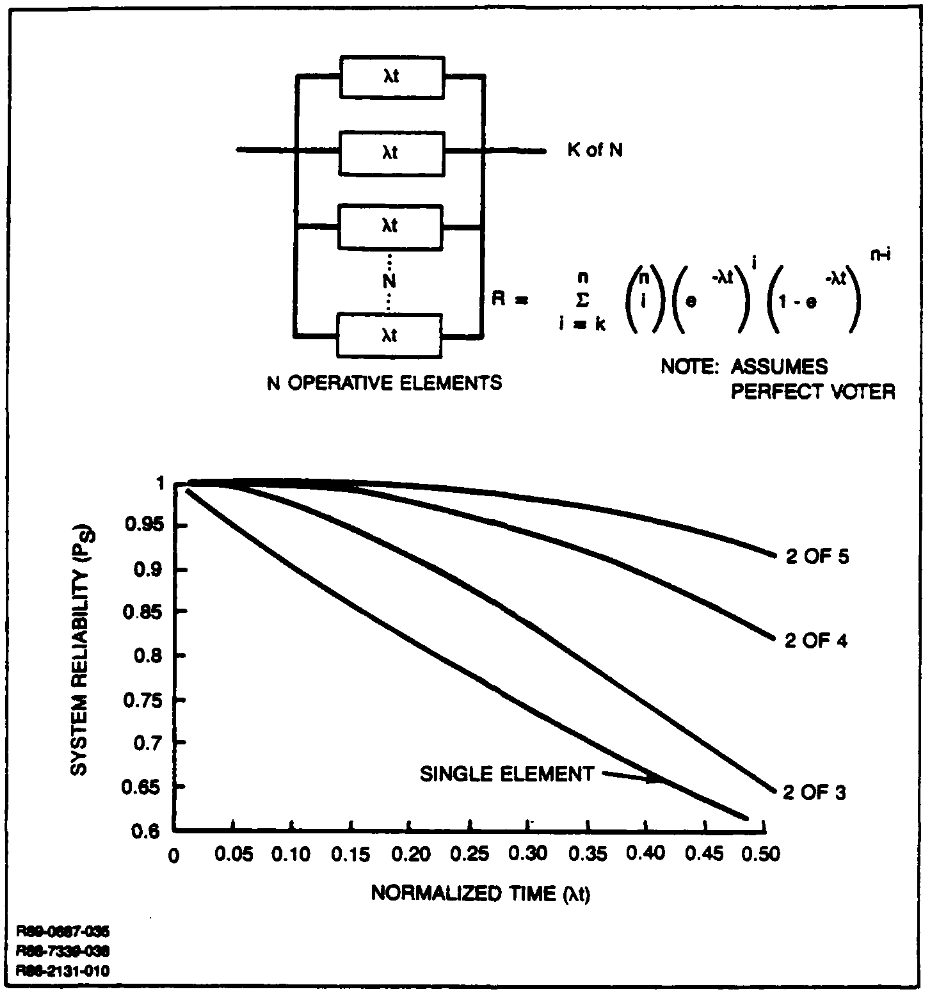


Figure 6-13. System Reliability for K out of N Redundant Configurations.

partitioning (see Subsection 5.3.6) of both system function and hardware must be emphasized so that the effect of a failure can be localized to the lowest hardware level at which reconfiguration is possible. The percentage of the faults to be detected and the accuracy of fault detection must be consistent with applicable reliability requirements. Several dynamic redundancy techniques are discussed in the following subsections.

6.1.5.1 Hybrid Redundancy

The application of dynamic redundancy techniques can eliminate a serious drawback in NMR-type configurations. Since the fault masking capability of an NMR design degrades rapidly as elements fail, the possibility exists for a collection of failed elements to out-vote the remaining healthy elements, thereby leading to premature system failure. However, replacing the failed elements dynamically with backup spare elements helps maintain the system reliability at a high level and eliminates many of the problems associated with voting ambiguity. This dynamic redundancy technique is often referred to as "hybrid" redundancy since it combines N-modular design techniques with those that implement backup sparing (see Subsection 6.1.5.3). The element(s) that are voted out by a majority of the NMR elements are replaced by backup spares that may be either hot, cold, or flexed (switched in periodically). Detailed tests can then be conducted on the suspected failed element(s) to confirm the failure. If the detailed tests corroborate the existence of the failure, the failed element(s) remain off-line pending corrective maintenance. However, when the detailed tests do not confirm the failure, the element(s) may be returned to the backup spares pool for use at a later time.

Figure 6-14 compares the system reliability of a hybrid TMR configuration as a function of individual module reliability and the number of spares. The illustration assumes perfect voter reliability and fault detection coverage inasmuch as these assumptions do not affect a comparison of the sensitivity of system reliability with the number of elements and element failure rates of the pooled spares concept. In Fig. 6-14a, the failure rate of the spare(s) is assumed to be equal to the on-line element failure rate. This corresponds to hybrid TMR with hot pooled spares. In Fig. 6-14b, the failure rate of each spare is assumed to be equal to 10% of the on-line element failure rate. This example represents a hybrid TMR with cold or flexing of spares. Comparing Fig. 6-14a and 6-14b also shows the range of module reliability values where relatively large increases in system reliability are obtainable relative to the number of pooled spares added. The comparison also shows where the use of cold pooled spares produces significantly increased system reliability over hot pooled spares. This assumes that the system can successfully fulfill its mission while a cold spare is brought on line.

Figure 6-14 also shows other values of module reliability below which single-string series (simplex) configurations have greater system reliabil-

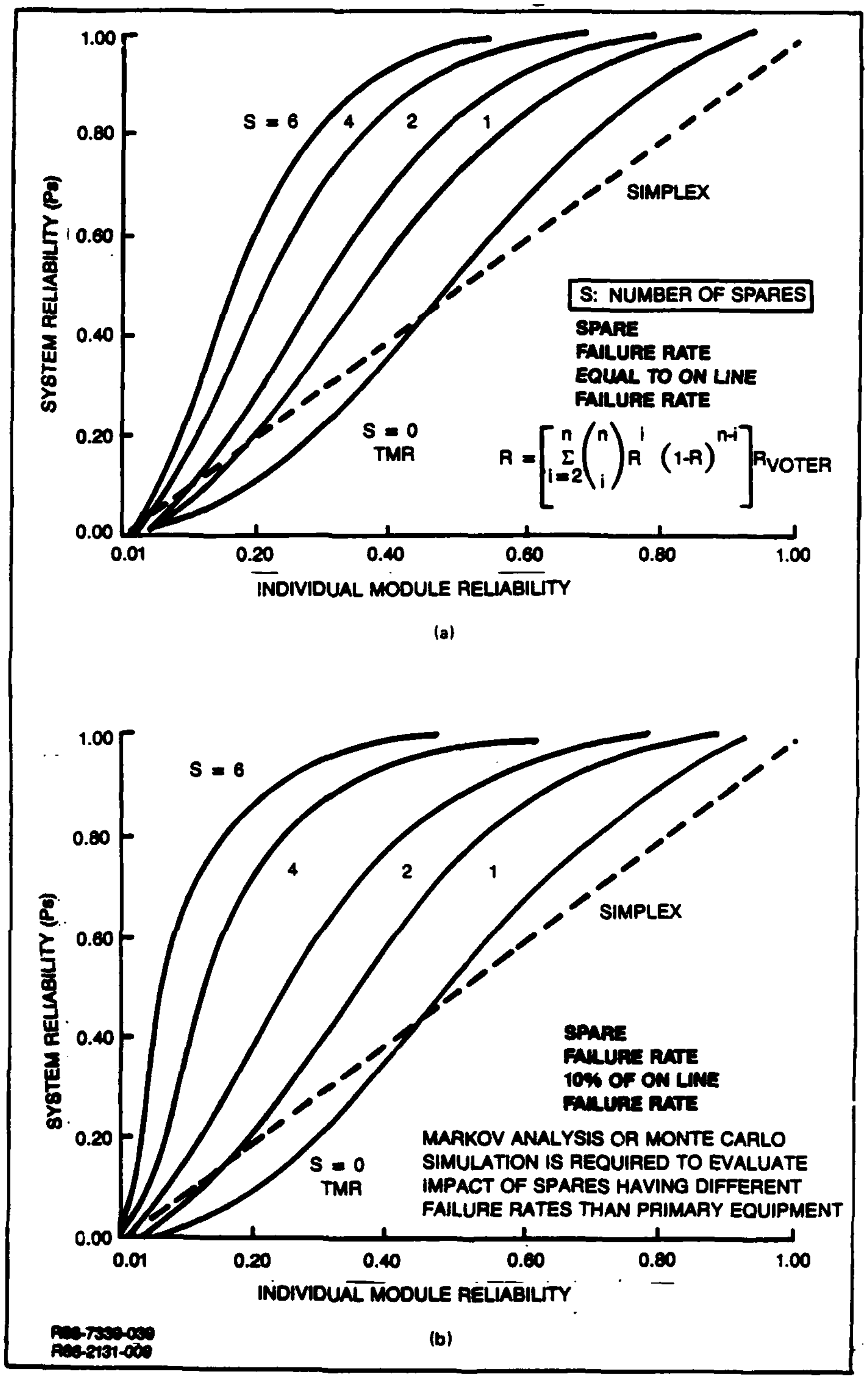


Figure 6-14. Hybrid TMR System Reliability as a Function of Individual Module Reliability & Number of Pooled Spares.

ity than selected TMR or hybrid configurations. These crossover points for TMR and hybrid system reliability are also found in many other commonly used redundancy configurations (see Fig 6-8). Technical managers should ensure that trade studies are performed when multiple redundancy strategies are being considered. There may be cases where prior studies have evolved a preferred system architecture for a generic class of C^3I systems. Nevertheless, alternate redundancy schemes should be explored if they promise benefits (i.e., less weight, improved power dissipation, schedule, or lower cost) and still satisfy functional and reliability requirements.

6.1.5.2 Adaptive Voting

Another form of dynamic redundancy involves an alteration to the voting scheme in response to failures. Disconnecting known bad modules from future votes eliminates the possibility of the failed modules outvoting the good modules. This technique is referred to as "adaptive voting", because the voting scheme is modified in response to equipment failures (Ref 3). Adaptive voting is most easily implemented under software control because hardware implementation of this technique tends to increase system complexity and part count.

Due to the hardware and software complexities of adaptive voting schemes, Monte-Carlo simulation techniques and/or Markov analysis is required to evaluate the reliability and availability of this type of fault tolerant design.

6.1.5.3 Pooled Spares

The use of pooled backup spares is not limited to applications with NMR-type configurations; rather, pooled spares can be used in a multitude of applications that include simplex, dual redundant, TMR, and large K of N configurations. The fact that the pooled spare modules are not dedicated to the performance of a particular function represents the key feature of this type of configuration. Depending on the application, the pooled spares can be cold, hot, or flexed (periodically checked out).

Cold spares are not operated until they are switched in; hence they exhibit lower failure rates than the operational modules (failure rates for dormant modules may be as low as 10% of the active module failure rate). Consequently, using cold pooled spares often results in higher system reliability than could be obtained by using hot spares. This approach often provides significant advantages in situations of long duration missions without maintenance (e.g., unmanned spacecraft). It may also result in fewer spares, lower power requirements, and reduced weight over a hot sparing strategy.

One disadvantage of using cold spares is the length of time it takes to bring a cold spare on line. Certain mission or safety-critical C^3I applications of cold sparing may result in an unacceptable risk of losing a critical system function. For example, if a second failure occurred prior to or during reconfiguration of TMR processors, outputs of the two remaining processors would disagree. This could result in losing critical data while attempting to distinguish which of the processors has failed. When considering the use of cold spares in a specific system, the time necessary for powering-up, initializing, self-checking, data transfer, and synchronization must be calculated. If the time required is unacceptable, configurations of hot or flexed spares should be considered, particularly in mission and safety-critical applications.

Another disadvantage of using cold spares is the possibility that the spare will not function when called upon. Spare modules are subject to many of the same environmental stresses (e.g., vibration, shock, and thermal cycling) that the operating modules experience. Consequently, the presence of latent faults may cause spare modules to fail when activated. Therefore, the following should be considered when devising pooled sparing strategies for mission and safety critical applications:

- Design soft turn-on circuitry (e.g., limiting inrush current) for equipment being considered as cold pooled spares
- Consider the use of hot standby equipment
- Consider flexing of the spares.

Hot pooled spares are modules or equipments that are powered and operating in a slave mode. They may be shadowing the operating elements, but their output is not being used or voted upon. The delay time to reconfigure is thus minimized and takeover by the slave is virtually instantaneous. The slave needs no updates because it is performing the same tasks as the primary elements. The disadvantages of using hot pooled spares are the increased probability of failure for long duration missions and the increased power and weight required to achieve the desired reliability.

Flexed spares are spare system elements that are exercised periodically and systematically. This process serves to expose latent faults in spare elements and greatly decreases the probability that the spare will not function when it is configured into the system. Flexing requires that the spare element be periodically powered up, a process that markedly increases the element duty cycle and can, in some cases, increase the element failure rate beyond that of a hot spare. Thus, individual system elements should be analyzed to assure that flexing will not degrade element reliability below acceptable limits.

The frequency at which the spares must be flexed is a function of element MTBF, size/complexity/module count of the C^3I system, and function/system criticality. In general, the period may range from seconds to several minutes, and should be a mere fraction of the total mission time.

6.1.5.4 Graceful Degradation

Graceful degradation is a design technique that utilizes extra hardware as part of the system's normal operating resources to ensure, with high probability of success, that an acceptable (minimum) performance level can be maintained in the presence of failures. The added hardware may raise system performance above minimum requirements; this enhanced performance continues as long as the excess hardware is not required to overcome failure effects. Potential failure modes that cause only a partial loss of functional capability may require lower levels of fault tolerance, thereby reducing hardware complexity and overall system cost. **The extra hardware used in gracefully degrading systems differs from standby redundant and hybrid redundant configurations in that it contributes to normal system performance and does not have to be switched in.**

Examples of gracefully degrading systems include large C^{3}I phased-array radar systems and distributed processing systems. A phased-array radar antenna typically contains a large number of transmit and receive elements. A small number (typically less than 5%) of randomly dispersed failures of these elements has a negligible effect on system performance, and additional failures can be compensated for by boosting transmitter power or receiver gain. An even larger number (typically less than 10%) of random element failures might be offset by the capability of the surviving elements to meet minimum acceptable system performance requirements (see Fig. 6-15) with a degraded detection capability. These antennas are adaptable to a deferred maintenance policy wherein failed elements need not be repaired after each mission. A second example of graceful degradation is a distributed data processor subsystem in which the network contains extra operating processors that provide additional throughput (see Subsection 6.4). If any processor fails, only the excess throughput capacity is lost. The number of extra processors to be included in the network can be selected to yield an allocated probability of maintaining at least minimal system functionality through the end of the mission.

Graceful degradation implies that element failures are unlikely to cause extensive secondary failures. Limiting secondary failures, i.e., fault containment, often requires careful design of the interconnections between adjacent and groups of adjacent phased array radar elements. **Technical managers should ensure that an FMECA is performed at a functional or hardware level so as to indicate the consequences of element failure(s) in a gracefully degrading system.** The level of detail in the FMECA should be consistent with that necessary to highlight design susceptibility to data contamination or secondary failure(s) so that corrective redesign activity can be aimed at containing the undesirable failure mode.

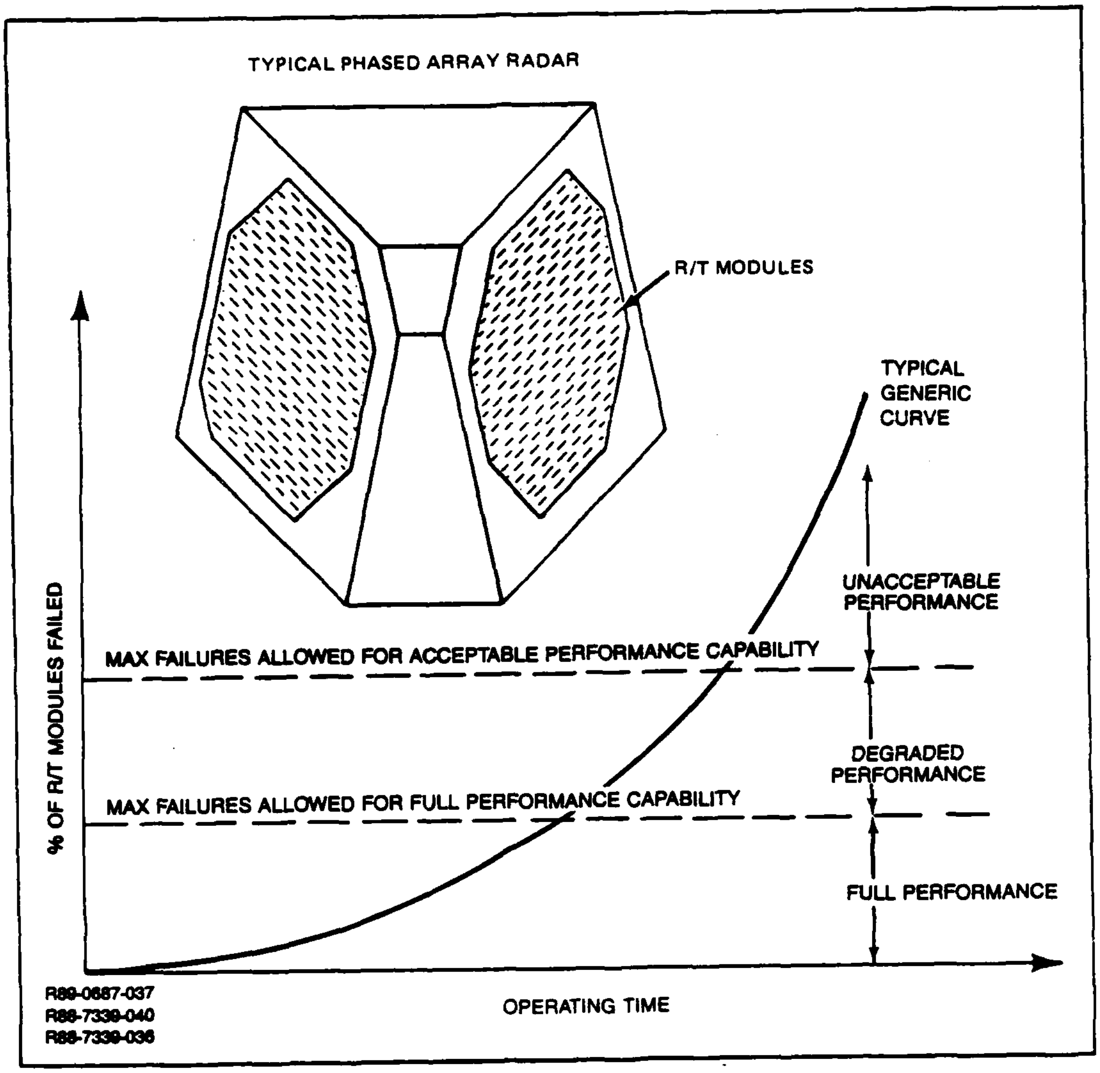


Figure 6-15. Graceful Degradation of Antenna Receive/Transmit Modules.

6.1.6 Hardware Redundancy Checklist Questions

a. Can fault avoidance techniques be used in lieu of redundancy to achieve the system requirements?

b. What system requirement has driven the decision to incorporate redundancy?

c. Has the dormant failure rate (if applicable) for standby redundant elements been considered in tradeoff analyses of active vs standby redundancy?

d. Has redundancy been considered for all mission and safety critical functions?

e. Has an effective fault detection and isolation scheme been developed and analyzed for all redundant hardware?

f. Where redundancy is used, has consideration been given to avoiding common mode failure situations that could disable all redundant paths?

g. Has a detailed FMECA been performed to uncover any susceptibility of failure propagation and to confirm FD/FI provisions?

h. Has the decision to incorporate redundancy been based on an analysis of the tradeoffs involved?

i. Have the penalties (i.e., increased maintenance, weight, volume, complexity, cost, spares, design/development time) associated with added redundancy been considered?

j. Has the reliability of switching devices needed to implement redundancy been considered in the reliability analysis?

k. Have the cost benefits of other reliability improvement techniques (e.g., parts derating, design simplification, environmental stress screening, etc) been considered prior to the decision to add redundant hardware?

l. Where the number of added redundant units exceeds the equivalent of triple or quadruple redundancy, has the diminishing incremental increase in system reliability been considered?

m. Has the level(s) of implementation of redundancy been selected with testability considerations in mind?

n. What alternate redundancy techniques have been identified that satisfy the allocated reliability requirement? Do these alternates result in lower system weight or cost?

o. Has the need to periodically check the health status of standby redundant elements been considered?

p. Has the use of dynamic redundancy and the pooling of spares been considered as an alternative to dedicated active or standby redundancy?

q. Has the length of time required to bring cold spares on line been considered in the analysis of standby redundant and pooled spare configurations?

r. Have the following approaches been considered when pooled spares are to be employed in mission and safety critical applications?

- Design soft turn-on circuitry for cold spares
- Operate with standby spares
- Operate with flexing of spares.

s. Has the increased probability of failure for hot spares been considered in cases where long mission durations are a factor?

t. Has the increased duty cycle (with resulting failure rate increases) of flexing of spares been considered?

u. Have fault containment provisions been incorporated to prevent secondary failures?

v. What is the hardware penalty for implementing the fault tolerance?

6.2 Software Fault Tolerance

Software fault tolerance is a term that applies both to software techniques used to deal with hardware faults, and software that is tolerant of imbedded faults. Both of these areas are discussed in this section. Software and hardware fault tolerance really are subsets of system fault tolerance. This system view is emphasized since fault tolerance functions in modern designs can be implemented in hardware and/or software. Therefore, fault tolerance functions are, in reality, system functions and it is most appropriate to deal with software fault tolerance as it relates to the entire system.

Software fault tolerance techniques provide mechanisms for complex systems to continue operation after a software fault occurs. The software fault may result from either a design/interaction fault or be induced by a hardware fault (Ref 4). Design/interaction faults are often detected during the operational phase of the system life cycle when a path in the program that contains the fault may be exercised for the first time or when the operator interacts with the system in a way that was not anticipated during system development. This occurs as a result of the complexity of modern C^3I systems. It is not unusual that fielded software is not exhaustively tested since testing all paths for all conditions is impractical. The approach utilized is to test extensively, not exhaustively. Therefore, due to incomplete testing during software development, some paths are not fully exercised. To minimize this occurrence, technical managers are cautioned to concentrate on requirements definition, code walk-throughs, and extensive unit tests. Hardware fault tolerance mechanisms that are incapable of dealing with a particular hardware fault (whether permanent or transient), may result in errors in software routines being executed, and system failure is common. This latter category of software fault is referred to as a hardware induced error, and it can result in problems with even well-tested software routines.

Techniques for implementing software fault tolerance range from the approach that uses single or multiple copies of identical software controlling the reconfiguration of similar redundant hardware elements, to systems where multiple copies of dissimilar software control the reconfiguration of redundant dissimilar hardware resources (see Subsection 6.2.1). Figure 6-16 provides a description of several software fault tolerance techniques that should be considered during software design and development. Figure 6-17 summarizes a number of error detection techniques that are commonly implemented in software.

Frequently, fault tolerant systems use a combination of software fault tolerance techniques. The choice of approach has major cost and schedule implications and should be made by technical managers only after a thorough analysis and evaluation of risks (see Subsection 4.1) and failure consequences (see Subsection 7.1) have been performed.

6.2.1 N-Version Hardware and Software Fault Tolerance Techniques

The implementation of fault tolerance can draw on a broad range of software techniques and often encompasses similar and/or dissimilar software and hardware. Figure 6-18 provides an evaluation of "N-version" fault tolerance techniques that are essentially a generalization of the N-version software technique to include both hardware and software. N-version programming is defined as the independent generation of two or more functionally equivalent programs from the same initial specification. Depending upon the particular C^3I system application, the costs associated with the implementation of these techniques varies from trivial to prohibitive.

TECHNIQUE	DESCRIPTION
WATCHDOG TIMER	SOFTWARE RESETS A HARDWARE COUNTDOWN COUNTER TO A PRESET VALUE BEFORE IT REACHES ZERO. IF THE COUNTER REACHES ZERO IT TRIGGERS AN INTERRUPT AFTER WHICH THE SYSTEM INVOKES A PREDETERMINED RECOVERY/SHUTDOWN PROCEDURE
COMMUNICATION TIME-OUT	SOFTWARE DETECTS A LAPSE IN COMMUNICATIONS BEYOND A SPECIFIED MAXIMUM TIME BETWEEN MESSAGES. SOFTWARE THEN ATTEMPTS TO RECOVER, OR TO ASSIGN OPERATION TO ALTERNATE HARDWARE COMMUNICATION RESOURCES
N-VERSION SOFTWARE	MULTIPLE COPIES OF IDENTICAL OPERATIONAL SOFTWARE ARE SIMULTANEOUSLY EXECUTED IN INDEPENDENT, IDENTICAL HARDWARE CHANNELS. RESULTS ARE COMPARED FOR A MAJORITY DECISION.
SOFTWARE DIVERSITY	MULTIPLE COPIES OF SOFTWARE ARE EXECUTED, EACH DESIGNED AND WRITTEN TO THE SAME SPECIFICATION, BY AN INDEPENDENT GROUP. THE DISSIMILAR SOFTWARE IS EXECUTED IN INDEPENDENT AND IDENTICAL HARDWARE RESOURCES. RESULTS ARE COMPARED FOR A MAJORITY DECISION
SOFTWARE RELOADS	PROVIDE THE ABILITY TO RELOAD ALL OR PORTIONS OF THE SOFTWARE TO FACILITATE A PARTIAL OR COMPLETE RECOVERY OF THE SYSTEM, OR TO ALLOW DYNAMIC RECONFIGURATION OF THE SYSTEM FOR OPTIMIZATION PURPOSES. THE SOFTWARE RELOAD TECHNIQUE CAN BE IMPLEMENTED AUTOMATICALLY AND/OR MANUALLY
BLOCK RECOVERY	A FUNCTIONAL SEGMENTATION OF OPERATIONAL SOFTWARE WHICH ALLOWS FOR CHECKS OF INTERMEDIATE RESULTS AND ALLOWS A VERIFICATION OR A REPEAT PROCESS OF LIMITED SECTIONS OF THE SOFTWARE
REPEAT PROCESSING	A SEGMENT OF OPERATIONAL SOFTWARE IS REPEATED AT LEAST THREE TIMES. THE RESULTS ARE COMPARED FOR CONSISTENCY, AND ONE SELECTION IS MADE
DATA FORMAT & SEQUENCE CHECKING	INPUT DATA ARE SCREENED FOR ADHERENCE TO DESIGNATED FORM AND CONTENT CRITERIA. ALL DATA TRANSFERS ARE CHECKED FOR PROPER SEQUENCE
TICKET CHECKS	SOFTWARE POSTS UNIQUE IDENTIFIERS TO SIGNIFY THE EVENTS OF ENTRY AND EXIT OF MAJOR MODULES. THE PROCESSING FLOW IS TRACKED AND THE ABILITY TO BACKTRACK TO THE POINT OF FAILURE IS PROVIDED. THIS TECHNIQUE USUALLY RESULTS IN A SIGNIFICANT SOFTWARE OVERHEAD PENALTY
INPUT CORRELATION	CAPABILITY PROVIDED BY SOFTWARE AND/OR HARDWARE TO SCREEN OUT DATA INCONSISTENCY BEFORE THE DATA ARE PASSED TO THE OPERATIONAL PROGRAM(S)
R89-0687-038 R88-7339-041	

Figure 6-16. Software Fault Tolerance Techniques. (Sheet 1 of 2)

TECHNIQUE	DESCRIPTION
CHECKSUMMING	THE CONTENTS OF SUCCESSIVE MEMORY LOCATIONS ARE SUMMED, USING A SIMPLE ARITHMETIC OPERATION. THE RESULTS ARE STORED FOR REFERENCE DURING FUTURE REPETITIONS OF THE SAME OPERATION.
REPLACEABLE ROM BASED SOFTWARE	SYSTEM HAS THE ABILITY TO RECEIVE ALTERNATE VERSIONS OF ROM-BASED SOFTWARE OR FIXES (PATCHES) FOR IDENTIFIED PROBLEMS. CONFIGURES ITS OPERATIONAL SOFTWARE FROM MEMORY STORAGE THAT CONTAINS THESE CHANGES.
PARITY	AN ADDITIONAL BIT IS ADDED TO THE WORD LENGTH USED. THE NUMBER OF 1's IN THE WORD IS COUNTED WHEN THE DATA IS STORED OR TRANSMITTED. THE STATE OF THE CHECK BIT IS SET TO MAKE THE TOTAL NUMBER IN THE WORD AN ODD NUMBER FOR ODD PARITY, OR AN EVEN NUMBER FOR EVEN PARITY. PARITY PROVIDES SINGLE-ERROR DETECTION, BUT CANNOT ISOLATE THE FAILED BIT IN THE WORD. ERRORS OCCURRING IN EVEN MULTIPLES WILL NOT BE DETECTED.
BLOCK PARITY	A PARITY SUM IS FORMED FOR BOTH ROWS (WORDS) AND COLUMNS OF A DATA BLOCK. ISOLATION TO A SINGLE BIT IS POSSIBLE WHEN BOTH ROW AND COLUMN PARITY SHOW A FAILURE INTERSECTION. MULTIPLE FAILURES CAN BE DETECTED BUT NOT ISOLATED.
ERROR DETECTION AND CORRECTION CODES R89-0687-039 R88-7339-041 (1/2)	SYSTEMATIC CODING OF TRANSMITTED DATA USING DISTINCT CLASSES OF CODES CONFIGURED TO DEAL WITH SPECIFIC TYPES OF ERRORS.

Figure 6-16. **Software Fault Tolerance Techniques.** **(Sheet 2 of 2)**

DETECTION TECHNIQUE	APPLICATION	IMPLEMENTATION LEVEL	ISOLATION CAPABILITY	RESPONSIVENESS	COMPLEXITY
VOTING/COMPARISON	ANALOG ELEMENTS, DIGITAL LOGIC	MODULE, FUNCTION, UNIT	FINE-TO-COARSE	HIGH	LOW TO MEDIUM
WRAP-AROUND	ANALOG AND DIGITAL ELEMENTS	MODULE, FUNCTION, UNIT	FINE-TO-MEDIUM	MEDIUM	LOW TO MEDIUM
PARITY	DIGITAL TRANSMISSION AND STORAGE	DIGITAL WORD	FINE	HIGH	LOW
CHECKSUM	DIGITAL TRANSMISSION AND STORAGE	DIGITAL WORD BLOCK	COARSE	HIGH	LOW
ERROR DETECTION/ CORRECTION CODES	DIGITAL TRANSMISSION	DIGITAL WORD	FINE	MEDIUM	MEDIUM
SYNCHRONIZATION	DIGITAL PROCESSES	FUNCTION, UNIT	COARSE	LOW	MEDIUM TO HIGH
WATCH-DOG TIMER	DIGITAL PROCESSES	FUNCTION	COARSE	LOW	LOW
DATA REASONABLENESS	ANALOG OR DIGITAL PROCESSES	FUNCTION	MEDIUM	HIGH	MEDIUM
ANALYTIC REDUNDANCY	ANALOG ELEMENTS OR PROCESSES	FUNCTION, UNIT	MEDIUM	MEDIUM	MEDIUM TO HIGH
DIAGNOSTIC SOFTWARE	DIGITAL PROCESSES	MODULE, FUNCTION, UNIT	FINE TO MEDIUM	LOW	HIGH
TOTALLY SELF CHECKING/FAULT SECURE NETWORKS	DIGITAL PROCESSES	DIGITAL WORD	FINE	HIGH	MEDIUM
R88-7339-042 R87-3537-012(T) R89-0687-040					

Figure 6-17 **Software Error Detection Techniques.**

TECHNIQUES	DESCRIPTION	DISADVANTAGES	ADVANTAGES
SIMPLEX HW & SW	SIMPLEX HARDWARE & SOFTWARE WITH NO FAULT TOLERANCE CAPABILITY	• NO RECOVERY FOR DESIGN OR TRANSIENT FAULTS	• LOW COST • RAPID DEVELOPMENT & IMPLEMENTATION
n CHANNELS OF HW/SW	EXECUTION OF IDENTICAL SOFTWARE IN REDUNDANT HARDWARE MODULES	• NO RECOVERY FOR DESIGN FAULTS • DECREASE IN SYSTEM THROUGHPUT	• REDUCES IMPACT OF TRANSIENT FAULTS • REDUCED SOFTWARE OVERHEAD
n CHANNELS OF K REDUNDANT HW & dSW	PARALLEL EXECUTION OF DISSIMILAR SOFTWARE IN n CHANNELS OF SIMILAR HARDWARE	• HIGH SOFTWARE OVERHEAD • NO RECOVERY FOR HARDWARE DESIGN FAULTS • DECREASE IN SYSTEM THROUGHPUT	• REDUCES IMPACT OF TRANSIENT FAULTS • REDUCES IMPACT OF SOFTWARE DESIGN FAULTS
n CHANNELS OF K REDUNDANT SW & dHW	PARALLEL EXECUTION OF IDENTICAL SOFTWARE IN n CHANNELS OF DISSIMILAR HARDWARE	• SEVERE HARDWARE OVERHEAD • DEVELOPMENT SCHEDULE AND COST IMPACT • NO RECOVERY FROM SOFTWARE DESIGN FAULTS	• REDUCES IMPACT OF TRANSIENT FAULTS • NO SOFTWARE OVERHEAD • REDUCES IMPACT OF HARDWARE DESIGN FAULTS
n CHANNELS OF K REDUNDANT dHW & dSW	PARALLEL EXECUTION OF DISSIMILAR SOFTWARE ON n CHANNELS OF DISSIMILAR HARDWARE	• SEVERE SOFTWARE OVERHEAD • SEVERE DEVELOPMENT SCHEDULE & COST IMPACT • DECREASE IN THROUGHPUT	• REDUCES IMPACT OF TRANSIENT FAULTS • REDUCES IMPACT OF HARDWARE & SOFTWARE DESIGN FAULTS

NOTE: SW = SOFTWARE; HW = HARDWARE; dHW = DISSIMILAR HARDWARE; dSW = DISSIMILAR SOFTWARE
R89-0687-041
R88-7339-045

Figure 6-18. N-Version Fault Tolerance Techniques.

A good example of a system with software fault tolerance can be found in the software implemented fault tolerant computer. High levels of reliability are achieved by having each iteration of a task executed independently by a number of redundant modules and then using a two out of three vote to select data for subsequent tasks. If all outputs are not identical, the system logs an error and the executive software attempts to isolate the faulty unit. In the software implemented fault tolerant computer, fault tolerance is achieved as much as possible by software routines as opposed to hardware. The software routines provide both error detection and correction, fault diagnosis, system reconfiguration, and they prevent the propagation of faults through the system. By checking for faults only at module interfaces, the software implemented fault tolerant system design distinguishes only between healthy and failed units and makes no assumption about the type of failure mode encountered.

Studies that have been conducted to investigate the cost-effectiveness of various software fault tolerance techniques indicate that any extensive fault tolerant design tends to significantly increase software development costs. To assure cost-effective software development, technical managers should make certain that only proven design procedures are used when implementing fault tolerant software techniques. In particular, since the largest contributors to software errors are correlated faults caused by improper or incorrect software requirements, technical managers should pay particular attention to software specification reviews and not be in a rush to generate code and that code not be "spaghetti code" which can not be maintained.

6.2.2 Error Detection Codes

Six specific error detection code types are discussed in the paragraphs that follow. Their ability to be extended to error correction is cited where appropriate to provide a more complete understanding of their characteristics and applications. The complexity and detection/correction capability of these code types are summarized in Fig. 6-19. When specific

CODE TYPE	CAPABILITIES		COMPLEXITY
	DETECTION	CORRECTION	
PARITY	ANY SINGLE-BIT ERROR. NO DOUBLE-BIT ERRORS SOME MULTIPLE, ADACENT, UNI-DIRECTIONAL ERRORS	NONE	LOW
HAMMING	ANY SINGLE-BIT ERROR ANY DOUBLE-BIT ERROR	SINGLE BIT	HIGH
M-OF-N	ANY SINGLE-BIT ERROR 1-OF-3 DOUBLE-BIT ERRORS ANY MULTIPLE ADJACENT UNI-DIRECTIONAL ERRORS	NONE	MEDIUM
AN	ANY SINGLE-BIT ERROR	SINGLE BIT	LOW
RESIDUE-M	ANY SINGLE-BIT ERROR	SINGLE B'T	MEDIUM
CYCLIC	SINGLE-BIT TO MULTIPLE, RANDOM BITS. BURST ERRORS.	SINGLE AND RANDOM MULTIPLE SINGLE BURST	MEDIUM TO HIGH

R88-0687-042
R88-7339-043 (T)

Figure 6-19. Properties of Error Detection/Correction Codes.

code types are examined in actual application, they may utilize the characteristics of more than one category, and error correction is also often incorporated in their design.

6.2.2.1 Parity Codes - Parity is the most basic coding technique. Its characteristics are used in the more advanced and complex types of linear separable codes. Bit-per-word parity has led to bit-per-byte (8-bits), interlaced parity, and chip-wide parity. In each case a parity bit (odd or even) is generated from the state of the bits assigned.

6.2.2.2 Hamming Codes - These are linear separable codes which use the parity of predesignated bit positions in a word as their basis. Each bit position in the information word is numbered:

$X_1X_2X_3X_4$............

Checkbit "one" is made to represent the parity of all bit positions whose binary equivalent would contain a "one" in the first column:

$C_3C_2C_1$

$X_1 = 0\ 0\ 1$

$X_2 = 0\ 1\ 0$

$X_3 = 0\ 1\ 1$

$X_4 = 1\ 0\ 0$

$C_1 = X_1$, X3, X5, X7...............

Checkbit "two" represents parity of all bit positions in column two, and check bit three, the parity of column three bit positions:

$C_2 = X_2, X_3, X_6, X_7$...............

$C_3 = X_4, X_5, X_6, X_7, X_{12}, X_{13}, X_{14}, X_{15}$............

By properly decoding the checkbits, an error in a single bit can be delineated. The construction of the code is further refined by incorporating the checkbits into the information word so that they may also be checked:

$C_1C_2X_3X_4X_5X_6X_7C_8$...............

Because any single error can be identified, it can also be corrected, which puts hamming code in the class of error-correcting code. In this form, often an overall parity bit is added, making it a single-error correcting, double-error detecting code.

6.2.2.3 M-of-N Codes - An M-of-N Code (m/n code) consists of n-number of bits in the code word in which m, and only m, bits are ones. If, for example, there are four bits available, the code "space" is sixteen words ($2^4 = 16$). A 2/4 m/n code restricts valid code words to those which contain two ones. There are six of these words: 0011, 0101, 0110, 1001, 1010, and 1100. In general, the number of code words available in any code space is the number of combinations of n bits taken m-at-a-time:

$$\binom{n}{m} = \frac{n!}{(n-m)!m!}$$

This produces a nonseparable code whose information input must be encoded to be represented by a valid code word, then decoded after it is received and its validity checked. Separable m/n codes can be formed by expanding the encoding logic and adding coding bits. This can simplify detection circuitry. The numbe. of code words in a code space is more restrictive than parity (parity yields eight code words from a sixteen-word code space). However, its error detection capability is greater than that of parity.

6.2.2.4 AN Codes - AN codes are the simplest of the arithmetic code types. They are formed by multiplying the data word by a number, called the "modulus." The modulus is chosen to be a number other than the base in which the data is expressed (base two for binary). For example, a 3N code simply multiplies the N-bits of information by three and transmits the result. When the code word is received, it is divided by three to confirm that the data is evenly divisible. Any remainder signifies that an error has occurred. By relating the remainder to the errors that occur, it is possible to designate and correct a single-bit error.

6.2.2.5 Residue Codes - Residue-m codes are generated to obtain separable arithmetic code types. The data word is multiplied by a modulus and the result concatenated with the original data word. The process is repeated by the receiving elements and the calculated result compared with the received residue.

6.2.2.6 Cyclic Codes - These are the most powerful codes for detecting and correcting random and burst errors. They are considered linear codes, because any cyclic (end-around) shift of a code word produces another valid code word. To construct the code, a polynomial representation of the data word is used. This "generator polynomial" is a primitive root of the coded word length and it completely and uniquely characterizes the code. The chosen root determines the cyclic code's detection capability.

The generator polynomial is expressed as an equation whose terms and coefficients are dependent upon the chosen root. For example:

$$G(X) = X^{16} + X^{12} + X^{5} + 1$$

A linear encoder/decoder is derived from the terms of G(X). Figure 6-20 is a typical block diagram of a cyclic encoder. Information bits, comprising words or blocks, are transmitted to the receiver and simultaneously fed to the encoder/decoder circuitry. After all bits have been shifted, the block check register contains the check bits, which then are transmitted. The data is again encoded by the receiver and its resultant check bits compared to those transmitted. A detectable error will result in a difference that can be used to designate the bits in error.

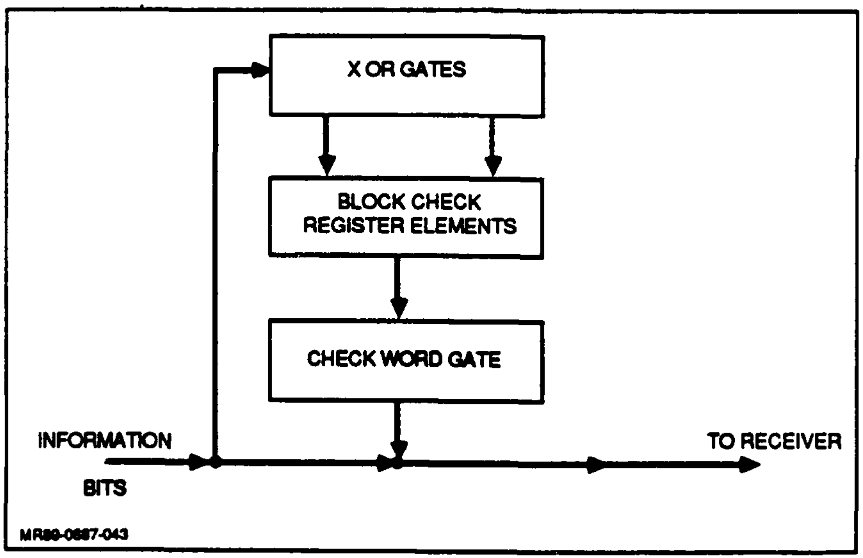


Figure 6-20. Cyclic Encoder Elements.

Common cyclic codes used in data communications are BSC (binary synchronous communication), BCH (Bose-Chaudhuri-Hoequenghem), Reed-Solomon, and fire codes. BSC is capable of detecting three or fewer independent errors and two bursts of length two. Capabilities of the code, as with all cyclic codes, depend on the specifics of the generator polynomial and message length. A BCH code, generally used in voice-grade channels, is capable of detecting four errors. The BCH code is the one most often used for detecting and correcting multiple, random errors.

Reed-Solomon and fire codes are the best known classes of block codes for dealing with multiple bursts of errors in code words. Detection and correction of two burst errors in a code block have been used.

6.2.3 Error Correction Codes

In many C^3I systems, it is imperative that data not be lost. These systems must have a built-in capability to correct data as well as to detect errors. Error correction codes (ECC) constitute a technique that has been used extensively to protect against errors occurring in systems. ECC is implemented by using a codeword of length N which consists of K bits of data and an additional P check (or parity) bits. As the codeword is read, the check bits are tested by an algorithm to determine if an error exists. In most cases, any bits that are in error can be identified and correction made by completing them. ECCs generally are limited to the detection of two bit errors and the correction of single bit errors in the codeword.

Since errors can occur singly, in random multiples, or in bursts due to timing inconsistencies, distinct classes of ECCs have been configured. More complex error patterns require more sophisticated coding techniques. ECC types can generally be assigned to one of the following categories:

- **Separable Code** - Contains two parts: the original data and the code bits. Decoding is performed by re-encoding the information bits and comparing the result with the received code bits. Parity and checksum codes are prime examples of separable codes. Linear separable codes divide the information word into bytes or fields of bits, each of which are encoded to form the check-bits

- **Non-Separable Code** - Forms an information word by translating or encoding the original data into a valid code word. The predetermined code pattern is examined, followed by decoding to extract the original information.
- **Arithmetic Code** - Generated by multiplying the information word by a number, or "modulus." After the data is transmitted it is divided by the modulus, and if a remainder exists a failure is noted
- Cyclic Code - Produced by a digital technique that uses linear feedback shift registers and "exclusive - OR" gates as adders. As the information bits are transmitted, an end-around shift produces a unique checkword which is sent after the information word. The receiver encodes the data as it is received and checks the result against the received check bits. Since the check bits are separate and distinct from the information bits, the cyclic code also can be termed a separable type code.

6.2.4 Software Fault Tolerance Checklist Questions

The following checklist questions should be helpful in guiding the software design process:

a. Are the software requirements properly defined? (Commitment from all parties to ensure full comprehension and agreement with defined requirements is mandatory.)

b. Has the selection of software algorithms been consistent with the prioritized reliability goals?

c. Have the system impacts of the selected algorithms been identified?

d. Have recovery algorithms been developed that correspond to signals from the fault detection algorithms?

e. Does the software have the capability to determine when recovery algorithms have failed so that a controlled system deactivation or transition to a degraded mode of operation can be effected?

f. Have simulations, modeling, and analyses been used to determine whether system software reliability and fault tolerance goals have been met?

g. Has an iterative process of software design refinement been established to facilitate the achievement of reliability goals?

h. Have error correction codes been considered for inclusion in the software?

i. Do the levels of software fault tolerance meet system failure resiliency criteria?

j. What is the software overhead penalty for implementing fault tolerance?

6.3 Fault Avoidance Techniques

A number of design techniques, commonly referred to as fault avoidance techniques, are available as cost-effective methods of increasing system reliability and decreasing maintenance requirements. Since these techniques serve to prevent, by construction, the occurrence of a fault, they should be examined to determine applicability and carefully evaluated in parallel with the development of the baseline fault tolerant design.

Typically, fault avoidance techniques include the following:

- Reduction of environmental stresses
- Use of military-grade piece parts
- Application of a stringent parts derating policy for new designs
- Imposition of environmental stress screening at the piece part and equipment levels
- Use of proven circuit design methods that assure high reliability.

These techniques are discussed briefly in the subsections that follow. A more comprehensive treatment of fault avoidance techniques may be found in MIL-HDBK-338.

6.3.1 Derating

Derating can be defined as the operation of an item at less severe stresses than those for which it is rated. Derating can be accomplished by either reducing stresses (i.e., applied voltage, temperature, vibration level, etc) or by increasing the strength of the part. In practice, the selection of a part of greater strength is usually the most practical approach. Derating has proven effective because the failure rate of most components decreases when the applied stress levels are decreased below the rated value.

As a general rule, derating should not be conservative to the point where costs rise excessively; neither should the derating criteria be so flexible as to render reliable part application ineffective. Optimum derating occurs at or below the point on the stress temperature curve where a rapid increase in failure rate is noted for a small increase in temperature or stress.

Comprehensive information on electrical and electronic device derating can be found in MIL-HDBK-338. Air Force derating requirements and guidelines can be found in Ref 27 and 28. Navy part application and derating requirements/guidelines can be found in Ref 29 to 31.

6.3.2 Environmental Stress Screening (ESS)

ESS is a test or series of tests specifically designed to disclose weak parts or uncover workmanship defects. This type of testing is widely used during the manufacture of electronic equipment because it can significantly reduce the negative effects of manufacturing, quality, and system process defects on field performance. The defects, commonly termed "latent defects," are traceable to poor workmanship (i.e., cold solder joints), out-of-control processes, or defective parts and assemblies. If left uncorrected, these latent defects can have a severe impact on the removal rates of hardware in the field, and result in reduced field reliability and system readiness.

Test conditions and procedures for ESS are typically designed to *stimulate* failures usually experienced in early field service, rather than to provide precise *simulation* of the operational life profile. Environmental stress tests (such as random vibration testing and thermal cycling tests) can be applied *in series*, rather than *in combination*, and should be applied to assembly levels for which they are most cost-effective.

Hardware that has not had ESS exposure can exhibit much higher removal rates during early or even sustained operational life than predicted or demonstrated reliability baseline values would indicate. Since defects are in effect attributes of specific equipments and not a function of the inherent design life, the application of ESS to such equipment under controlled conditions can result in significant improvements in field performance. **Since ESS offers significant potential for improving field reliability, and thereby the availability of C^{3}I systems, widespread application to new systems is recommended.** This is particularly true for fault tolerant system developments, since higher reliability levels can be achieved by ESS.

MIL-STD-2164(EC) defines the approach and method to be used for ESS testing so that latent defects can be located and eliminated before the equipment is accepted. The standard requires that the constituents and sequence of the ESS test be as shown in Fig. 6-21. In general, generic test levels and durations that are included in design requirements documents should be analyzed by designers so they can take into consideration all static and dynamic loads associated with operation, accelerated environmental testing, storage, shipping, and ESS acceptance testing.

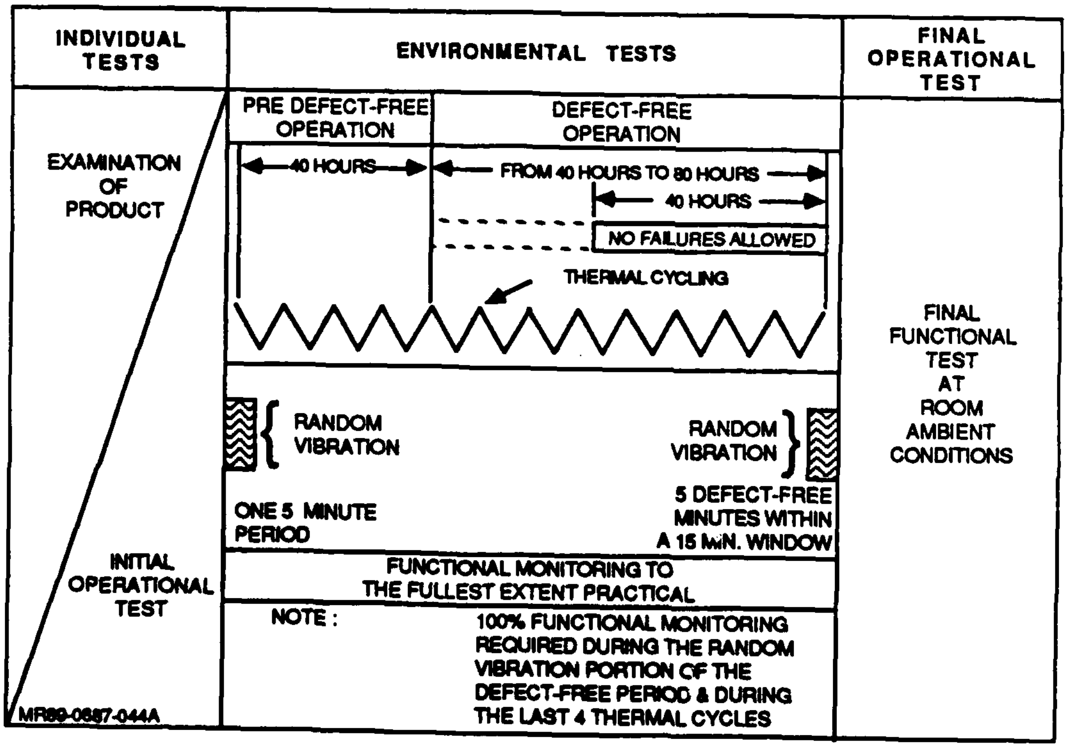

INDIVIDUAL TESTS
ENVIRONMENTAL TESTS
FINAL OPERATIONAL TEST
EXAMINATION OF PRODUCT
PRE DEFECT-FREE OPERATION
DEFECT-FREE OPERATION
40 HOURS
FROM 40 HOURS TO 80 HOURS
40 HOURS
NO FAILURES ALLOWED
THERMAL CYCLING
RANDOM VIBRATION
RANDOM VIBRATION
ONE 5 MINUTE PERIOD
5 DEFECT-FREE MINUTES WITHIN A 15 MIN. WINDOW
FUNCTIONAL MONITORING TO THE FULLEST EXTENT PRACTICAL
NOTE : 100% FUNCTIONAL MONITORING REQUIRED DURING THE RANDOM VIBRATION PORTION OF THE DEFECT-FREE PERIOD & DURING THE LAST 4 THERMAL CYCLES
INITIAL OPERATIONAL TEST
FINAL FUNCTIONAL TEST AT ROOM AMBIENT CONDITIONS
MR89-0887-044A

Figure 6-21. Environmental Stress Screening Test Constituents.

6.3.3 Part Selection and Control

A crucial part of the design implementation process is the specification, selection, application, and control of component parts to be used in the system. The part selection process for fault tolerant systems is similar to that for other systems. However, since from a reliability viewpoint the system can be no stronger than the components from which it is built, the technical manager should be familiar with part selection and control techniques.

Criteria and guidelines for the selection and control of a broad range of electronic components are contained in MIL-HDBK-338 together with a detailed discussion of quality and screening tests. Wherever possible, the designer should endeavor to use standard electronic parts in the equipment design inasmuch as these parts have proven to be more reliable than nonstandard parts. Their use will help improve overall system reliability and help minimize LCC. Nonstandard parts, materials, and processes should be avoided if possible; when used, they should be interchangeable with a standard equivalent and be as reliable as a standard equivalent.

6.3.4 Reliable Circuit Design

Technical managers should assure that proven circuit design methods that ensure high reliability are used in the system design. Fault tolerant systems are particularly dependent upon reliable circuit design since it is imperative that added design complexity does not significantly increase the system's series failure rate. In general, successful fault tolerant designs will evolve from consideration of the following reliability design criteria:

- Design simplification
- Use of standard parts
- Component derating
- Use of transient and overstress protection
- Degradation of part operating characteristics
- Minimized design errors
- Adherence to fundamental design limitations.

Design simplification ranks with parts selection and component derating as an effective means of increasing reliability. Technical managers should carefully determine whether all features and circuits in the design are needed to perform the intended functions. Design simplicity contributes to optimal reliability by making system success depend upon fewer components, with a resultant decrease in potential failures. When attempting to simplify a design, the technical manager must exercise caution so that:

- Higher stresses or unusual performance requirements are not imposed on system components
- Nonstandard or unproven parts are not used in attempts to replace multiple parts with a single part capable of performing multiple functions
- System fault tolerant features are not compromised.

Design errors that include deficiencies that cause performance overstress or R/M/T problems can be prevented by implementing an informal procedure to check circuit designs. Technical managers should make certain that early circuit designs are checked by experienced designers and reliability specialists for reliability design errors. The results of these reviews should be communicated directly to the designer along with suggestions on how the deficiency can be eliminated.

A more detailed treatment of reliability design criteria and reliable circuit design techniques may be found in MIL-HDBK-338 and Ref 8 and 9.

6.3.5 Environmental Stresses

Technical managers and designers must understand the environment and its potential effects on system operation and reliability. Selecting designs that can withstand environmental effects and using techniques that serve to alter or control the environment are important ingredients in the fault tolerant design process. By selecting designs or materials that can withstand the operational environment, the designer can constrain the system's complexity since components that serve to control the environment need not be added. MIL-HDBK-338 contains a great deal of information on environmental design and specific techniques for assuring reliability in the presence of various environmental factors.

Because temperature changes influence the physical properties of almost all known materials, the impact of operating temperature on electronic equipment reliability is an extremely important design consideration. Figure 6-22 illustrates how the normalized total system failure rate of a typical avionic system varies as a function of operating temperature (Ref 10). The mix of devices present in this system included signal processors, 1750A processors, computer mass memory devices, remote terminals, power supplies, data processors, and displays that were implemented using state-of-the-art LSI, VLSI, and VHSIC technology. The knee of the curve represents the range of interest for the reliability analyst, since operating at temperatures in this region will likely result in lower system LCC (reduced failures, corrective maintenance, and spares). Operating at lower temperatures might significantly increase the size and complexity

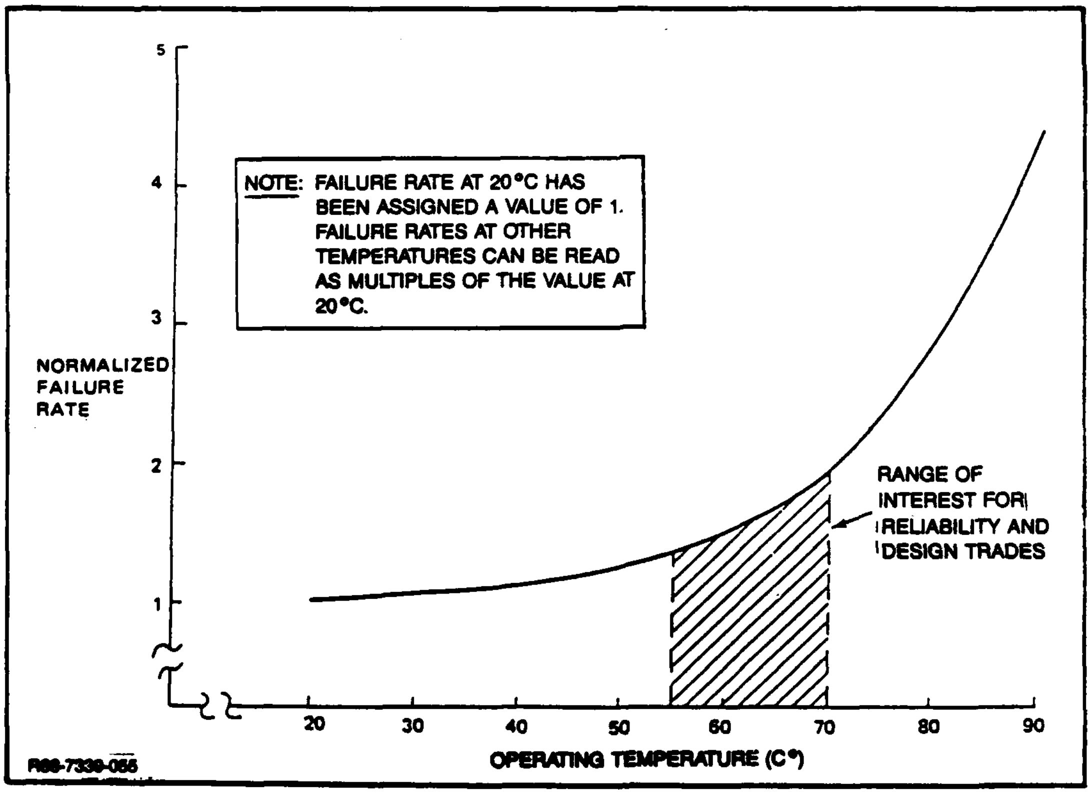


Figure 6-22. Normalized Total System Failure Rate vs Junction Temperature.

of the equipment cooling system with associated decreased total system serial reliability and increased total system weight.

Technical managers should assure that each component is studied to determine if a substitute is available that will generate less heat, or if the component can be located or positioned to minimize the thermal effect on other components. In general, the appropriate arrangement of components, when coupled with efficiently integrated heat removal techniques, can significantly improve system reliability. The preferred method of evaluating thermal effects on the reliability of electronic equipment is to use the MIL-HDBK-217 parts stress analysis technique. This method establishes the maximum allowable temperature for each part in a circuit and includes consideration of the failure rate allocated to the component and overall equipment reliability requirements.

Technical managers should make certain that the design process adequately addresses thermal design considerations, because they are as important as circuit design in obtaining the necessary performance and reliability characteristics from the equipment. The fine points of thermal design constitute an engineering discipline unto themselves and are discussed in detail in Ref 11. The potential effects of thermal, shock, and vibration environmental stresses are discussed in Ref 1 along with the effects of moisture, radiation, sand, and atmospheric pressure.

6.3.6 Fault Avoidance Checklist Questions

The following checklist items provide the technical manager with a convenient means to determine whether appropriate fault avoidance design techniques and procedures were used in the fault tolerant system design:

a. Has an up-to-date preferred parts list been used in the parts selection process?

b. Have part application guidelines been developed and adhered to during parts selection?

c. Does reliability data/experience support the use of nonstandard parts?

d. Have parts been reviewed for proper application?

e. Have the dominant failure modes of each part been considered prior to parts selection?

f. Do parts used in the design meet environmental requirements for temperature, shock, vibration, humidity, etc?

g. Do the part derating guidelines correspond to specification requirements?

h. Have the part characteristic operating variations due to aging, temperature change, etc, been analyzed?

i. Have thermal analyses been performed to determine component operating temperature?

j. Do internal cooling provisions limit internal temperature rise?

k. Are high power dissipation components properly heat sinked?

l. If water or conditioned air is to be used as a heat sink, have components been properly sealed and shielded to prevent moisture problems?

m. Do conducting surfaces, surface coatings, paints, adhesives, and conformal coating materials have good thermal conducting properties?

n. Have performance tests been conducted at temperature extremes to assure circuit stability over the full range of operating temperatures?

o. Are components whose failure rates are sensitive to temperature located away from heat flow paths, power supplies, and other high heat dissipation components?

p. Have vibration/shock analyses been performed to assure structural integrity and determine resonant frequencies that may be experienced in the operational environment?

q. Do cables/harnesses/wires have sufficient slack to prevent stresses due to thermal changes and vibration/shock?

r. Has environmental stress screening been considered for (or performed on) parts, components, subassemblies, assemblies or equipment to remove latent defects?

s. Have proven circuit design methods that ensure high reliability been employed?

t. Has the design been reviewed for possible simplification?

u. Have circuit designs been checked for reliability design errors by experienced reliability specialists?

v. Has the impact of increasing/decreasing equipment operating temperature on failure rate been considered?

w. Has a MIL-HDBK-217 parts stress analysis been performed on the equipment?

x. Have life tests or reliability tests of critical components/subassemblies been conducted?

y. Has the use of limited life items been kept to a minimum?

z. Has the fanout of gates been limited to a small number so as to decrease power dissipation?

6.4 Distributed Processing Architectures

Architecture describes how a processing system is implemented and how it operates in terms of both its hardware configuration and software task functions. Distributed processing divides a system into separate computing resources to share a task load, to provide access to computational services, or to enhance fault tolerance. These characteristics are often used in conjunction with each other to satisfy the system requirements and concepts. System elements that are commonly divided are:

- Processing Elements

  Use of multiple processors, each of which performs differing tasks as part of a larger function

  - Multiple processors each performing the same task to speed computation, to provide fault tolerant redundancy, or to extend computing facilities to multiple users
- System Control
  - Application level of operating systems and executive routines to direct and control processor operation
- Data Base
  - Physically or logically partitioning a data base which may be replicated for sharing or independent access
- Physical Location
  - Geographic dispersion to provide user access or to reduce vulnerability to physical damage.

Distributed processing systems may evolve from simpler, centralized systems or may be designed and built for specific applications. When a digital system is initially configured or expanded, specific capabilities, functions and tasks for the system to perform are defined. Designers select and interconnect the hardware processors and interface components to mechanize the system. Operating software, including interface protocols and executive controllers, are written and installed to activate the system.

Basically, there are three primary characteristics that describe a distributed processing architecture:

- Processing element partitioning
- Interconnection of processing elements, commonly called network topology, and the protocol used to control message traffic
- Operating system partitioning and distribution among processing resources.

6.4.1 Processing Element Partitioning

The most basic (and earliest) form of a computer resource is a single central processing unit that is used to do all the required tasks of a system. This centralized system (see Fig. 6-23) performs all of the input/output (I/O) control and conversion and all the computation required by many and varied system tasks. When the capacity of the single resource

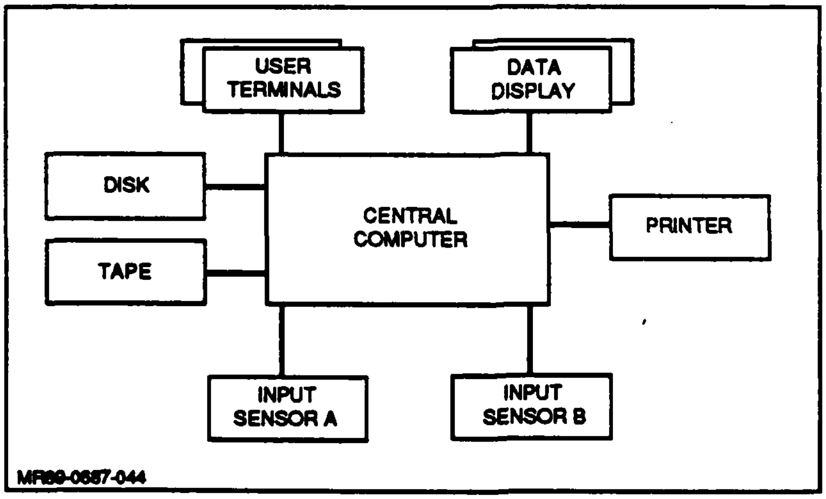


Figure 6-23. Centralized Processing System Architecture.

is exceeded, a second machine typically is installed with communication links to the first machine. This is the first partitioning level which yields a "horizontal" distribution (see Fig. 6-24) if the machines are executing the same tasks, or a "federation" of machines if they are not. Adding processing units internally within a machine is a related development characterized as multiprocessing. This developmental path leads to parallel processing, array processing, and their associated structures as shown in Fig. 6-25. They are considered to be of a class other than that

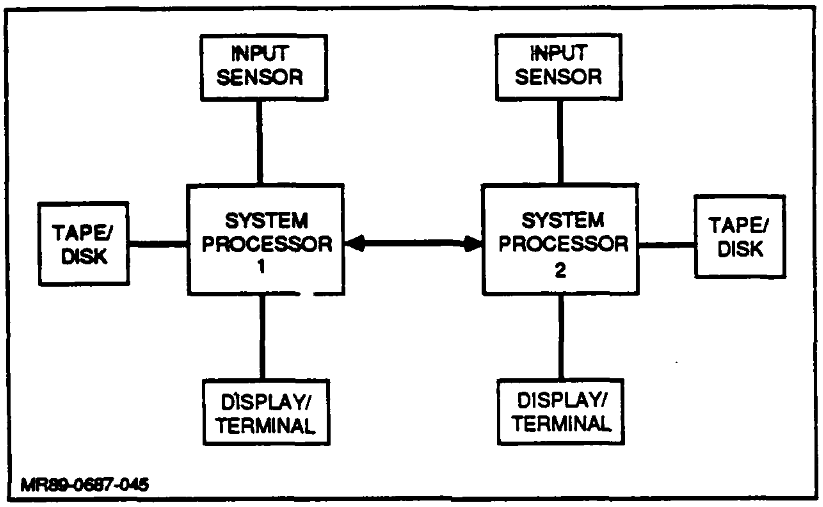


Figure 6-24. Horizontal Distribution or Federated System.

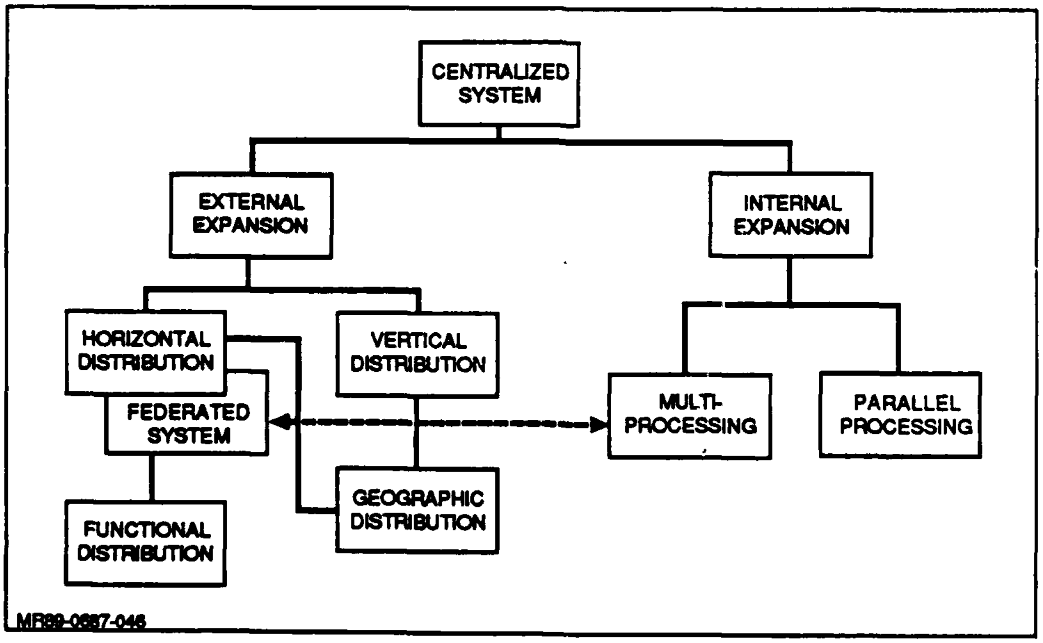


Figure 6-25. Distributed Processing Development.

addressed here that contain physically separated, independent machines interconnected by wired networks. The two classes overlap in systems that are configured to be contained in "integrated avionics racks" which commonly incorporate multiple processor, memory, and I/O modules interconnected by back-plane busses internal to the avionics rack.

Examples of the horizontal or federated structures occur in both ground installations and aircraft systems. The ground station often applies two (or more) mainframes or "supermini's" with a serial channel connecting the machines. Each may provide common capabilities for terminal operation and data base access as well as unique functions for data analysis, simulation or graphics output.

To conserve processing capability an airborne system usually will be configured to execute differing tasks. The interface between machines may be a parallel channel with status and command control lines. Software executive routines are designed to assume a subset of the failed computer's processing tasks. This affords a degraded back-up capability to perform the tasks most critical to the performance of the mission phase.

A centralized system can also be expanded in an "inside-out" manner to produce a vertically distributed form of architecture as shown in Fig. 6-26. Commonly it consists of a central computer, which may be a mainframe or "supermini" with external satellite processors, each connected directly to the mainframe by a separate data bus. The satellite proces-

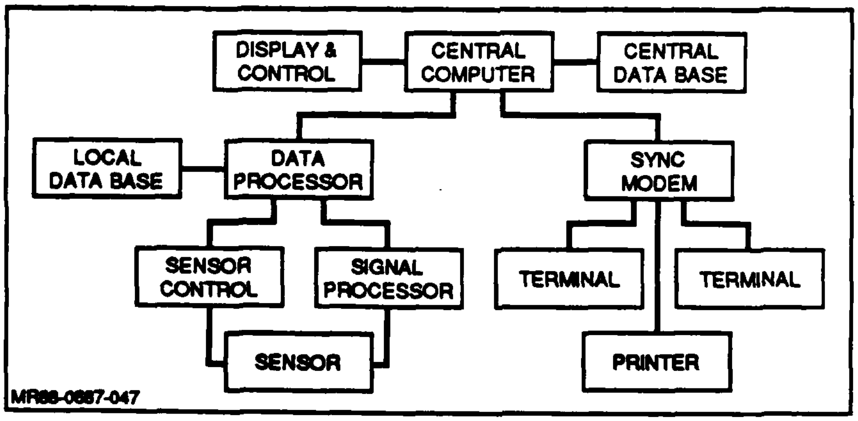


Figure 6-26. Vertical Distribution Architecture.

sors may perform identical tasks or serve as data "concentrators" for numerous input and output functions. They also may perform differing, specialized computational tasks as preprocessors for the central machine. In both designs, the expansion serves to reduce the computational, timing, and I/O loading of the centralized machine while augmenting and enhancing the total system capability. Fault tolerance is improved to the extent that satellite processors can assume the tasks of the failed system component. The central computer tasks are vulnerable to single-point failure. By combining the features of the horizontal or federated system with the vertical structure, a level of task redundancy can be implemented on the centralized computer level.

Processor and component miniaturization has enabled an additional level of system partitioning. It is now physically possible to install individual processors that are functionally dedicated to specific system functions. This "functionally distributed" system shown in Fig. 6-27 is most common in airborne systems. Processors embedded in subsystems such as inertial navigation, air data, radar sensors and displays each perform the subsystem I/O, data computation and control execution. System integration is achieved by data interchange between processors, usually by way of a multiplexed data bus. Fault tolerance is improved, since the system is less vulnerable to the single-point failures of the centralized computer. The effect of losing a subsystem and its associated data can be reduced by incorporating routines in alternate system processors which will produce similar, if not as precise, data. Back-up subsystems or instruments

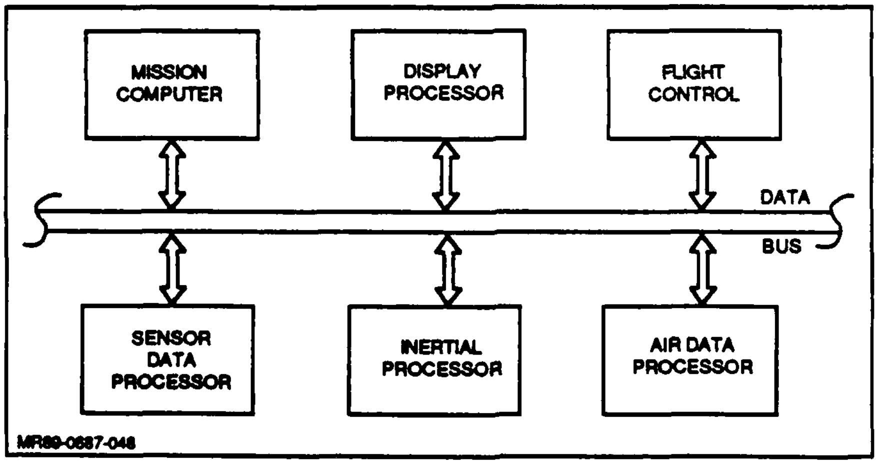


Figure 6-27. Functional Distribution Architecture.

can also be incorporated to provide similar, but coarser, information to the system.

The processors in any of the described structures are often in dispersed locations. For example, on an airborne distributed system for military aircraft, subsystem processors may be mounted in separate equipment bays to minimize vulnerability to battle damage. Some sensors have restrictions on where they can be located. Placing the conversion equipment and processors nearby reduces noise on interface signals. To minimize battle damage or to provide convenience in user access to processing facilities, the elements, or whole structures, are often geographically dispersed. Loss of a system "node" does not halt system operation since alternate working facilities can be used as long as network paths are available and the processing tasks to be performed can be assumed by the working elements.

6.4.2 Network Topology and Protocol

To form an operable system, a multiplicity of processors, terminals, displays, sensors and storage media must be interconnected. Interfacing these elements involves three basic forms of topology:

- Star Network
- Ring Network
- Multidrop Bus.

6.4.2.1 Star Network Topology

The Star Network is the earliest form of network topology. It is applied to a centralized system with all processing and control tasks executed in a single, central machine. Each system element is connected directly to the processor through individual I/O ports. Interface to each element is often dissimilar in operation. It may consist of standard bus protocols, such as RS-232 or IEEE 488 or non-standard, uniquely designed configurations. These may be full or half duplex, use "handshake" request/acknowledge signals and transmit either bit serial or parallel words. I/O control of each individual channel resides in the central computer operational software. Failure of any element in the network is not recoverable, so the entire system may be inoperable if the central computer fails.

As with processing structures, the Star Network (Fig. 6-28) can be expanded in an "inside-out" manner, using "front-end" or preprocessors, data concentrators and multiplexors. This expands the network's interface capability, using the central machine's individual I/O channels to connect to a satellite processor, rather than to an individual device. This is the "Tree" network, presented in Fig. 6-29, which is associated with vertical processor distribution. The network is still vulnerable to single-point failures which can disable a network section or the entire system.

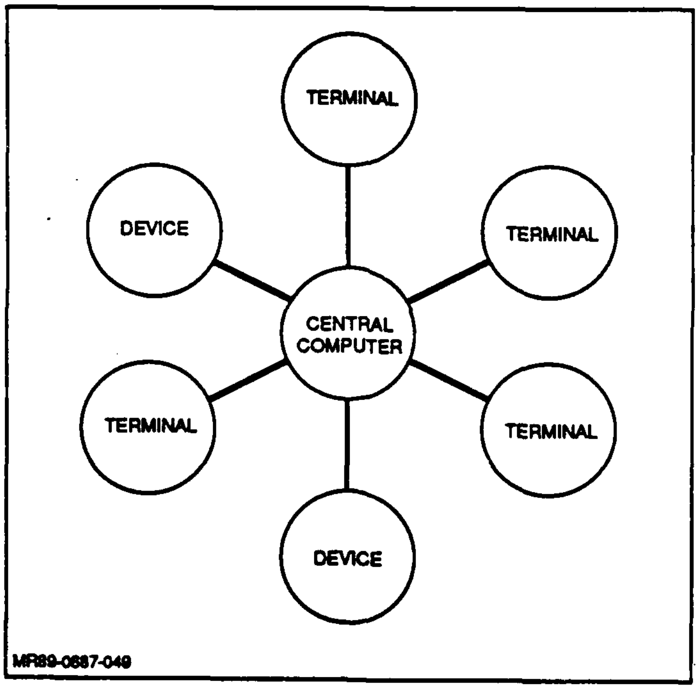


Figure 6-28. Star Network Topology.

6.4.2.2 Fully Connected Network Topology - To overcome the drawback of single point failure of the vertical processor distribution, a horizontal distribution of the central computer can be made, with interfaces extending to the first level of front-end processors. Adding additional processors and connecting each with every other processor develops the topology of the Fully Connected Network as shown in Fig. 6-30. Since an independent, buffered I/O channel must be included and dedicated to

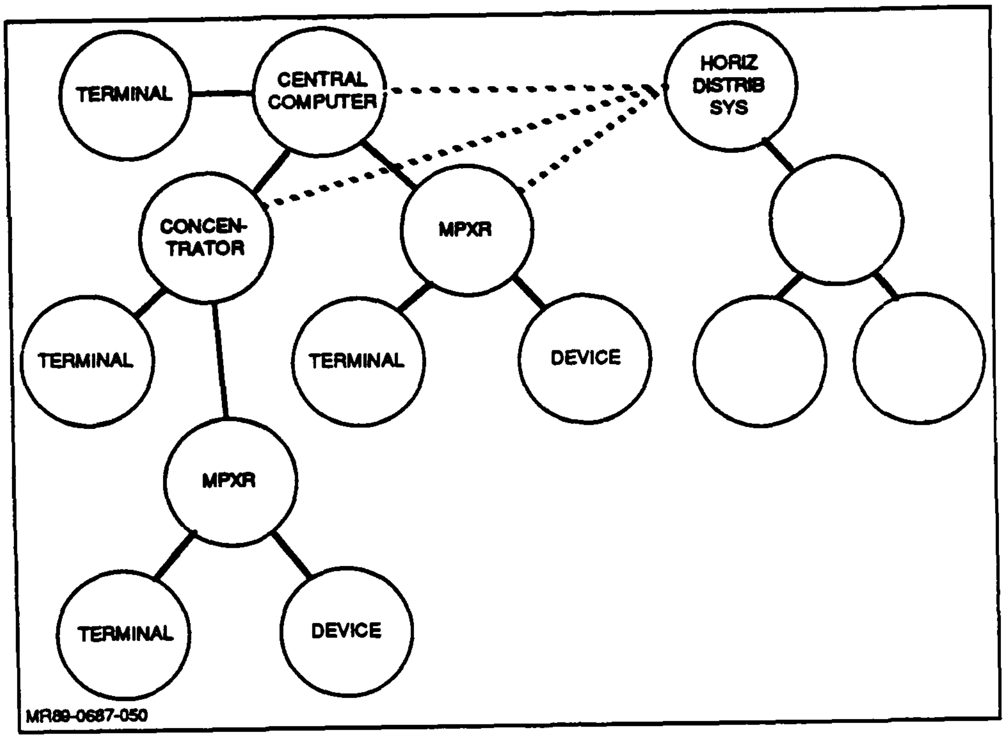


Figure 6-29. Tree Network Topology.

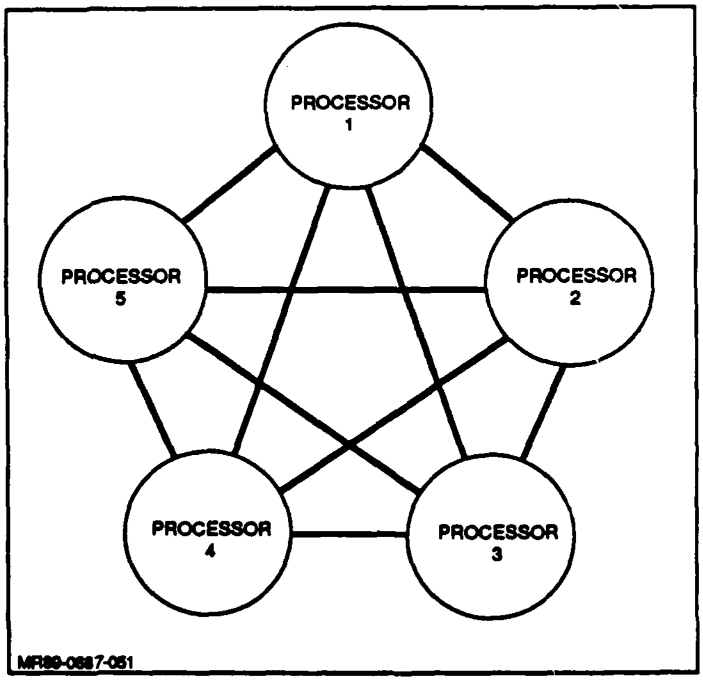


Figure 6-30. Fully Connected Architecture.

each processing element, the network is cumbersome and expensive for anything but a limited number of nodes. Loss of a node does not disable the system, so a higher level of fault tolerance is available. This is useful on a critical system such as digital flight control where multiple (usually up to four) processors are executing the same tasks. A functionally distributed system can also be fully connected, provided that there are a limited number of separate processing functions.

As the number of functions and processors increases, the number of interface channels required for the Fully Connected Network increases. To curtail the complexity and expense of multiple, dedicated interfaces, a Ring Bus Network (see Fig. 6-31) has evolved. Each processor in the ring is connected to one adjacent processor, so a single interface element is needed for a single ring with serial transmission in one direction. Without direct connactions, the Ring Bus protocol enables each processor to communicate with every other processor on the ring. The terminal which initiates transmission contains a "token" message which designates it as the originator. Each terminal on the ring sequentially receives and retransmits the message, including the intended recipient of the message. When the sender receives its own transmitted message, it removes the

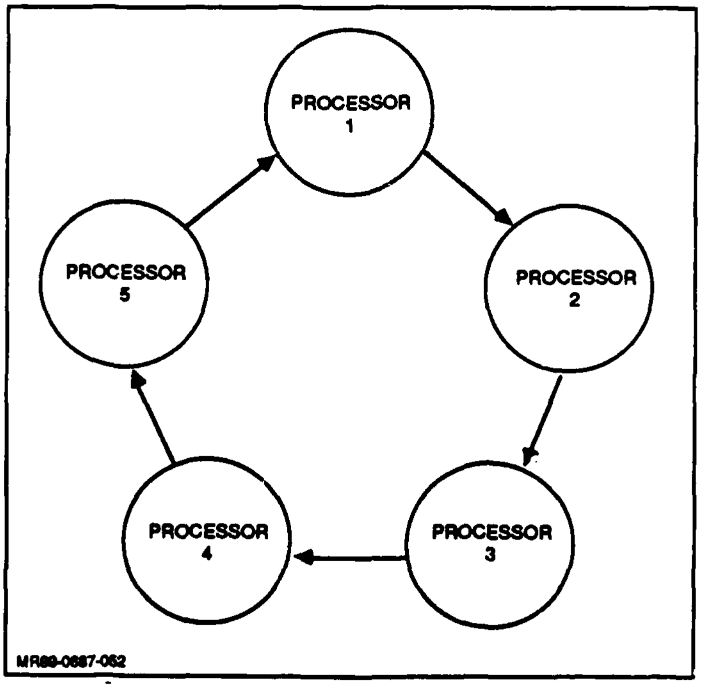


Figure 6-31. Ring Bus Architecture.

message from the ring. Then it transmits the token to the next station, which enables that terminal to transmit its message. A dual ring (see Fig. 6-32) is often implemented to prevent the failure of one node from interrupting the ring. The second ring transmits data in the direction opposite to the first ring and requires that a second interface element be installed in each node. This provides not only a second, standby, data path, but is often used to isolate a failed node by "looping back" a message received on the primary ring and re-transmitting it on the secondary ring (see Fig. 6-33). Bus delays are increased appreciably because the message must be received and transmitted twice by all but the end-most terminals. This can be considered part of a degraded mode of operation which excludes the functions of the failed node.

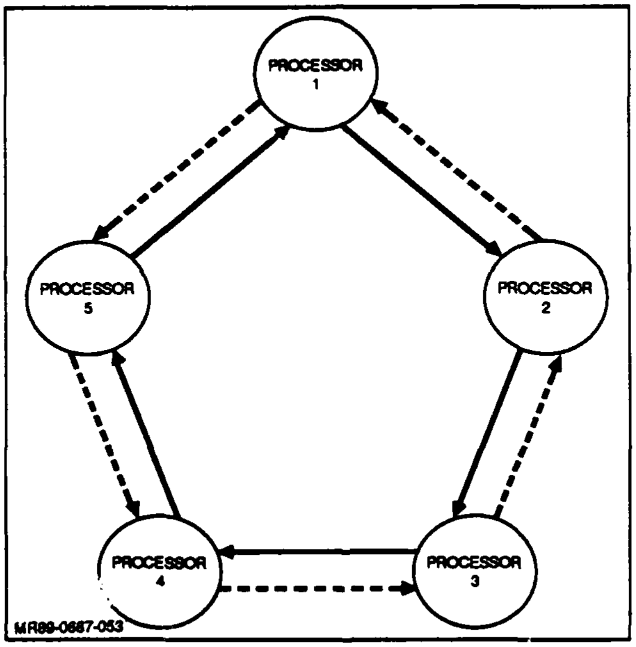


Figure 6-32. Dual Redundant Ring Bus.

6.4.2.3 Multidrop Bus Network Topology - In a broadcast-type bus, the third basic form of network topology, the transmitting station is heard by all of the other stations on the bus. This is commonly called the Multidrop Bus since the terminals are joined into a continuous bus medium, either optical or electrical. Three different protocols are used: command/response, token passing and carrier sense multiple access/collision detect (CSMA/CD). Each is used to maximize the operational characteristics of systems in which it is applied.

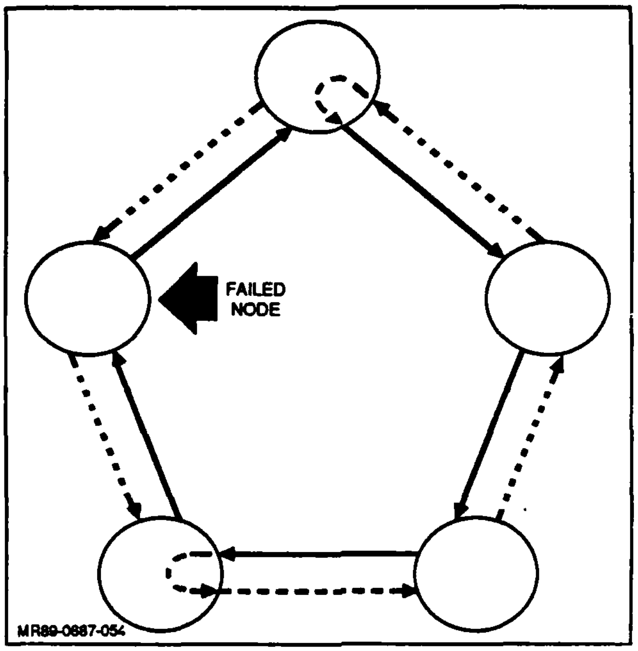


Figure 6-33. Dual Redundant Bus with Loop Back.

The MIL-STD-1553 Multiplex Data Bus is the prime example of the command/response protocol. It provides structured, synchronous control over all terminals on the bus. One terminal is designated as the bus controller and a second as a back-up controller. The bus controller initiates all remote terminal (RT) operations and data transfers by issuing command messages. The command message contains the RT address to which the command is directed, identifies the data to be transmitted or received, or designates an operational mode to be executed. The RT responds with a status word that acknowledges receipt of the command and then executes the required action.

Numerous features of the Multiplex Data Bus provide for fault tolerant operation. With a backup controller, failure of the primary bus controller can occur and the backup controller will automatically assume control. This provides a degraded mode standby redundancy, since the bus continues to operate, but the functions the primary bus control processor provides are unavailable. Multiplex bus systems are also most often configured with dual-redundant buses; that is, one bus provides the primary means of communication and the second bus is used as a standby redundant path. Bus controllers and RT interfaces are designed to utilize either bus for transmission upon command of the bus controller. Use of the RT status word provides a positive response to commands and transactions on the bus. If a command is not received or executed, the bus controller is apprised by the absence of the status word. In addition, the status word format contains coded results of RT self-test routines, so the bus controller can monitor the system's operational status. Finally, each word contains parity and is checked for proper sync character, normal coding, and correct number of bits. Depending upon the error and message type, the transmission is ignored and a message error bit is set in the status word.

The Linear Token-Passing Data Bus (LTPB), can be characterized as a serial multiplex data bus with circulating bus control. The terminal which possesses the token message, like the Ring Bus, initiates the transmission of its messages. Since the physical medium of the bus is not a connected ring, a logical ring is superimposed on the system. This establishes the order of terminals to which the token is passed. The station with the token is granted access to the physical medium for a predetermined maximum time. When the time has expired, or the terminal has sent all of its messages, it forwards the token to the next member of the logical ring. Messages are prioritized in the system to minimize message latency. If timers associated with low priority messages have expired because of an increase in traffic of higher priority, the station forwards the token to its successor.

LTPB failure prevention, detection, and reaction are implemented using bus redundancy, message frame check sequences, station management status messages, token passing protocol and station admittance timing. Dual redundant buses are applied as they are in the 1553 Multiplex Data Bus. A primary bus is used until a response failure is detected through a token passing message or a system monitor message. The alternate bus is then used by the token holder for message transmission. Station management status messages are used to report specific failures similar to the 1553 Data Bus status word. For the detection of transmission errors, the token frame and message frame formats provide check sequence fields of 8 bits and 16 bits, respectively. Cyclic redundancy checks, using specified generator polynomials of a cyclic code, are produced in the transmission of a message. The receiver contains the same logic circuitry to regenerate the check pattern over the bits transmitted. It then compares its results with the transmitted check sequence fields, enabling it to detect individual and burst errors which occurred in transmission.

More unique to the LTPB are the bus activity and ring admittance monitors. When a station passes a token, it verifies that bus activity resumes. If activity is not detected, it tries to pass the token again. If the second attempt fails, the station increments the destination address and transmits the token. This process continues until a successor is found or the destination address increments to the address of the local station. With dual-redundant buses, the token transmission can be attempted on the alternate bus prior to incrementing the address. A ring admittance timer, in conjunction with the bus activity timer, periodically enables a station to be admitted to a logical ring. They also serve to initialize the bus upon loss of the token or power-up.

The widely-used Ethernet commercial bus is the most common application of the CSMA/CD Multidrop Bus. It is designed to accommodate a large number of low-duty-cycle devices. There are no bus controllers or token holders, so it is the least structured of the protocols discussed here. The physical medium is a single coaxial cable which may be extended to 2500 meters with the use of repeaters. Multiple stations coupled to the medium (multiple access) are designed to monitor the bus for message activity. This is the "carrier sense" or "listen-before-talk" part of the convention. The message format contains both the source and destination address, so the station can identify and accept the messages directed to it. If the station has a message to be sent, it waits until there is no message traffic, then proceeds to the transmit. The station monitors its own transmission (listen-while-talk) and if it detects an alternation in the bits sent, it concludes that a transmission "collision" has occurred. The station will continue to transmit 32 to 48 additional bits, after which it will cease transmission for a random period based on a "slot" time of a predetermined number of bits. Retransmission of the message is then attempted. If collision continues to occur, the system is designed to make a finite number (usually 16) of attempts before relinquishing the bus. Errors in the transmitted message are detected by applying cyclic redundancy checks as described under the LTPB formats. The message frame format incorporates a 32-bit field for transmitting the generated check bits. Depending on the cyclic code used, it is possible to detect and correct single bit and burst errors received by the destination terminal.

6.5 Level of Implementation of Fault Tolerance

The level at which fault tolerance is implemented in the system design depends upon two key points:

- Level of detail (i.e., circuit, component, subsystem, subroutine, etc) to which the system can be decomposed
- Required level of fault tolerance.

Since fault tolerance is incorporated in both hardware and software (see Fig. 6-34), tradeoff analyses (Ref 12) are required to determine the optimum levels of implementation. These tradeoffs should consider the associated maintainability and testability implications and should determine if the redundancy is best applied at the chip level, board level, subsystem level, system level, or any combination thereof (Ref 13). Because of the diversity of C^{3}I applications, system requirements, and weight/power/volume/cost constraints, no general guideline as to the best level of re-

IMPLEMENTATION LEVEL	EXAMPLE
HARDWARE:	
PART	MICRO ELECTRONIC CIRCUIT, TRANSISTOR
CIRCUIT	LOGIC ARRAY, FLIP FLOPS
FUNCTIONAL	ADDERS, COUNTERS
SUBASSEMBLY	ARITHMETIC UNIT, MEMORY, CPU
EQUIPMENT	COMPUTER, GYRO, ACCELEROMETER
SUBSYSTEM	RADAR, COMMUNICATIONS
SYSTEM	RECONNAISSANCE SPACECRAFT
SOFTWARE:	
APPLICATION PROGRAM	KALMAN FILTER SOLUTIONS, TRAJECTORY MANAGEMENT
LOCAL OPERATING SYSTEM	SYSTEM EXECUTIVE, MSDOS
GLOBAL OPERATING SYSTEM	KERNEL EXECUTIVE, UNIX
R89-0687-066 R86-7339-068	

Figure 6-34. Levels of Implementation of Fault Tolerance.

dundancy can be provided. However, this subsection provides the technical manager with a discussion of many of the key aspects that enter into the decision and tradeoff process to determine the appropriate implementation level.

The evaluation of fault tolerant strategies for possible incorporation in the design must take into account the following:

a. What types of failures must be tolerated, and what is the probability of occurrence of each failure mode?

b. What are the effects and costs associated with these failure modes?

c. What recovery methods are available?

d. How much additional hardware and/or software is needed to provide the fault tolerance? (Ref 14)

From a system integration perspective, the incorporation of fault tolerance at low levels (e.g., part, circuit, application program, etc) may provide better control of the fault tolerance (i.e., redundancy management); however, system performance may favor the incorporation of fault tolerance at a higher level (e.g., subsystem, weapon replaceable assembly (WRA), line replaceable assembly (LRU), etc). For critical applications, fault tolerance may be achieved by functional partitioning and/or design diversity/ replication (Ref 15). By functionally partitioning the system, the effect of a component failure is localized to the subfunction level and does not adversely impact system functionality, although some degradation may result. Typically, functional partitioning results in a proliferation of hardware elements and is feasible where weight/power/volume are not constraints.

Design diversity can be defined as the development of two or more systems aimed at delivering the same service through independent designs (Ref 15). Design diversity/replication typically is implemented using multiple independent channels (see Subsection 6.1) which may be composed of independently designed hardware/software elements. By incorporating multiple channels in a system that uses VLSI or VHSIC devices, two (or more) channels may be placed on the same chip, or different chips (i.e., using either the same or dissimilar hardware) can be used for each channel. By putting multiple channels on a single chip, the channels can be configured as a series of "self-checking" pairs.

The effect of "common mode" failures on the on-chip redundant elements is dramatic and must be considered since it can nullify the advantages of multiple channels. A common mode failure is a nonrandom event, usually time- or stress-dependent, that is caused by a latent manufacturing defect, a design flaw, or a susceptibility to an unanticipated environment. These types of mechanisms have the undesirable property of having a high likelihood of showing up in multiple like equipment during a small time-interval. If redundancy is incorporated via multiple like hardware, common-mode failures must be avoided.

A study performed by one of the VHSIC manufacturers concluded that on-chip redundancy may be impractical (Ref 16). This conclusion must, however, be related to the predominant failure mode of the integrated circuit. If complete chip functionality is lost most of the time the chip fails (e.g., VCC shorts to ground, hermetic seal leakage, voltage spikes, or thermal runaway), then redundancy built into the chip may not be practical. However, if most of the failure modes affect a few localized outputs, then the benefits of having the circuit duplicated on the chip are apparent in that the addition of the redundancy can be accomplished with minimal impact on size and weight. Another consideration relates to the relative failure rates of on-chip interconnections vs external chip connections. Therefore, when deciding whether to incorporate redundancy at the chip level, it is important to first become familiar with the physics of the device failure and the percent contribution of chip failure modes.

In safety-related applications the use of design diversity/replication can result in the use of two or more independently designed and developed VLSI/VHSIC circuits that are functionally compatible (Ref 15). Furthermore, as VLSI and VHSIC circuits become more complex with more and more functions incorporated on a single chip, designers must consider the application of fault tolerance at the chip level, since the rapid increase in component count per VLSI/VHSIC devices has the potential to offset the increase in reliability of a single component (Ref 14). Hence, although these devices hold promise of increased reliability, the designer must consider the consequences of a fault and make appropriate use of fault tolerance techniques.

7 - R/M/T EVALUATION & TRADEOFF ANALYSES

This section provides the technical manager with the background necessary to evaluate and conduct R/M/T tradeoff analyses on fault tolerant C^3I designs. Two aspects of fault tolerant system evaluation are discussed. The qualitative (i.e.,Pass/Fail) aspect of system evaluation examines the design to assure that it includes the necessary defenses against faults. The quantitative aspect of system evaluation determines whether measures of dependable operation (e.g., reliability, probability of success, availability, LCC, etc) satisfy the system requirements. Included is a discussion of:

- Failure mode and effects analysis
- R/M/T evaluation models
- Operational readiness
- Logistic resource analyses
- Mission effectiveness
- Life cycle cost.

7.1 Failure Mode & Effects Analysis (FMEA)

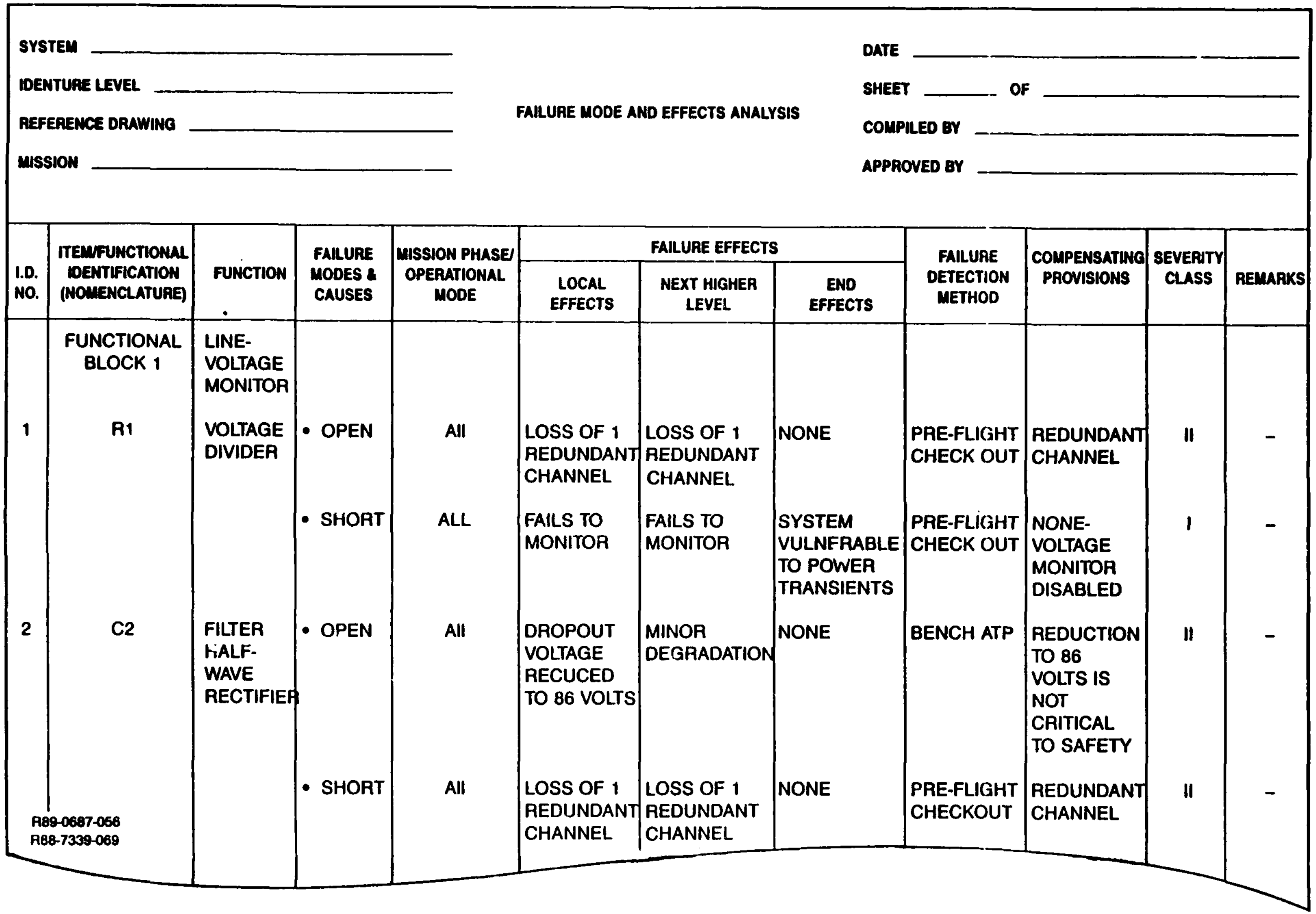
A FMEA is a procedure by which each potential failure mode in a system is analyzed to determine the results or effects on the system and to classify each potential failure mode according to its severity. The FMEA is a powerful tool in determining whether fault tolerant designs meet applicable reliability requirements (e.g., fail operational/fail safe, failure detection, incorporate provisions to prevent fault propagation, etc). Potential design weaknesses can be identified by analyzing engineering schematics and mission/operational rules to systematically identify the likely modes of failure, the possible effects of each failure (which may be different for each life/mission profile phase), and the criticality of each effect on safety, readiness, mission success, and the demand for maintenance/logistic support.

The FMEA utilizes inductive logic in a "bottom up" approach. Beginning at the lowest level in the system hierarchy, the analyst traces up through the system hierarchy to determine the effect each failure mode would have on system performance. The FMEA provides:

- A method by which the design engineer can select a design with a high probability of operational success
- An assessment of a design's fault tolerant capability
- A basis for design and location of performance monitoring, fault sensing devices, and other automatic test equipment
- Design engineering with a documented method for assessing the effect of failure modes on the system's operational success.
- Early visibility of system interface problems
- A list of possible failures which can be ranked according to their effects and probabilities of occurrence
- Identification of single failure points critical to mission success or crew safety
- Early criteria for test planning
- Quantitative and uniformly formatted data input to the reliability prediction, assessment, and safety models
- A tool which helps to evaluate proposed design, operational, or procedural changes and their impacts on mission success or crew safety.

FMEA activity should be initiated at the system level during the Concept Exploration phase where only functional failure modes may be identified because only limited design definition may be available. As greater design and mission definitions become available during later program phases, the analysis can be expanded to successively more detailed levels and ultimately, if required, to the piece-part level. The FMEA should be updated to reflect design changes since the existence of completed up-to-date FMEAs is a major consideration at design reviews.

Figure 7-1 is an example of a typical FMEA worksheet. For fault tolerant systems the analyst should give particular attention to identifying system level failure effects, failure detection methods, and compensating provisions for postulated failure modes. If the fault tolerant design is



SYSTEM ____________ IDENTURE LEVEL ____________ REFERENCE DRAWING ____________ MISSION ____________

FAILURE MODE AND EFFECTS ANALYSIS

DATE ____________ SHEET ______ OF ____________ COMPILED BY ____________ APPROVED BY ____________

I.D. NO.	ITEM/FUNCTIONAL IDENTIFICATION (NOMENCLATURE)	FUNCTION	FAILURE MODES & CAUSES	MISSION PHASE/ OPERATIONAL MODE	FAILURE EFFECTS			FAILURE DETECTION METHOD	COMPENSATING PROVISIONS	SEVERITY CLASS	REMARKS
					LOCAL EFFECTS	NEXT HIGHER LEVEL	END EFFECTS				
	FUNCTIONAL BLOCK 1	LINE-VOLTAGE MONITOR									
1	R1	VOLTAGE DIVIDER	• OPEN	All	LOSS OF 1 REDUNDANT CHANNEL	LOSS OF 1 REDUNDANT CHANNEL	NONE	PRE-FLIGHT CHECK OUT	REDUNDANT CHANNEL	II	-
			• SHORT	ALL	FAILS TO MONITOR	FAILS TO MONITOR	SYSTEM VULNFRABLE TO POWER TRANSIENTS	PRE-FLIGHT CHECK OUT	NONE-VOLTAGE MONITOR DISABLED	I	-
2	C2	FILTER HALF-WAVE RECTIFIER	• OPEN	All	DROPOUT VOLTAGE RECUCED TO 86 VOLTS	MINOR DEGRADATION	NONE	BENCH ATP	REDUCTION TO 86 VOLTS IS NOT CRITICAL TO SAFETY	II	-
			• SHORT	All	LOSS OF 1 REDUNDANT CHANNEL	LOSS OF 1 REDUNDANT CHANNEL	NONE	PRE-FLIGHT CHECKOUT	REDUNDANT CHANNEL	II	-

R89-0687-056
R88-7339-069

Figure 7-1. Typical FMEA Worksheet Format (MIL-STD-1629).

found to be inadequate, then alternate failure detection methods and/or the inclusion of additional compensating provisions should be considered.

A comprehensive discussion of FMEAs can be found in MIL-STD-1629.

7.2 R/M/T Evaluation Models

The design of fault tolerant C^{3}I systems must be supported by continual assessment of the system's ability to meet R/M/T requirements. The reliability math models discussed in Section 6 are appropriate for the analysis of relatively simple systems and/or where the consequences of imperfect fault protection coverage are not critical. Since current and future C^{3}I systems utilize extensive redundancy, complex fault management, fault recovery, and reconfiguration techniques, the trend towards these ultrareliable fault tolerant systems has necessitated the development of sophisticated R/M/T evaluation tools.

In the past, insufficient redundancy was considered the major source of system unreliability, and imperfect fault protection coverage was deemed a second-order effect. With increases in the complexity and sophistication of C^{3}I state-of-the-art and more stringent fault-tolerance requirements, the above two sources have achieved at least parity, if not complete role-reversal. Thus, any evaluation model must properly account for the effects of imperfect fault protection coverage.

Many of the evaluation tools used in the past are no longer adequate to deal with the high reliability levels and the complex fault handling mechanizations of today's systems. Monte Carlo simulation of the total system can be used in some complex cases where Markov Analysis or other analytical models are not flexible enough or are too complex to use. Such simulation has a calculable degree of error associated with it, and may at times require a large number of trials to obtain statistical confidence in the results. Caution should be taken when using any evaluation tool. Where possible, more than one tool should be used independently so that results obtained can be crosschecked to provide confidence in the result.

Simple analytical models based on series-parallel combinatorials (see Section 6) can adequately predict reliability when exhaustion of redundancy is the principal driver. The models assume that the system is composed of a series or a parallel arrangement of each of its independent constituent elements. The models either ignore the effects of imperfect fault protection coverage or, at best, treat the fault protection coverage probability as a "multiplier effect." This type of model fails to accurately or even adequately predict reliability when system failure dynamics and recovery are of consequence. The models cannot handle situations where a sequence of failures is important, where the failure is transient or intermittent, or where the response to failure (i.e., detection, isolation, recovery, reconfiguration) is imperfect.

7.2.1 Markov Analysis

Since fault-tolerant systems can operate in many different modes as a consequence of failure and failure management, reliability and availability modeling can be quite complex. Failures that occur may or may not be detected, and those that are detected may not result in correct isolation and reconfiguration of the system's resources. System reliability or availability figures of merit must be determined by considering that different failures and imperfect fault protection coverage decisions have their own impact. Therefore, current models for evaluating fault tolerant systems emphasize Markov methods. The Markov analysis relies on the notions of "system state" and "state transition" (Fig. 7-2). A state is a unique description of the system's operational status, usually characterized by the number of remaining (unfailed) constituent components.

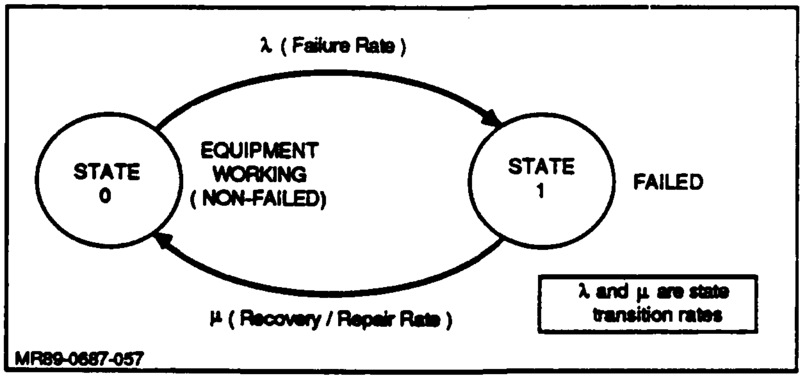

λ (Failure Rate)
STATE 0
EQUIPMENT WORKING (NON-FAILED)
STATE 1
FAILED
μ (Recovery / Repair Rate)
λ and μ are state transition rates
MR89-0687-057

Figure 7-2. Simple Markov Model.

Thus, for example, a system comprising three active and two standby units may have:

- 3 operating, 2 spares operational
- 3 operating, 1 spare operational, 1 failed
- 2 operating, 1 spare operational, 2 failed.

as some of the states that describe the system's status at any moment in time as a consequence of component failures. As mission time passes, the system goes from state to state by virtue of component failure and recovery. These passages are called state transitions and the rate at which the system goes from one state to another is called the transition rate. These transition rates are a function of the system's constituent component failure rates and failure-handling characteristics. Thus, the probability of the system being in any given state at some specified time can be determined from the initial conditions at the start of the mission and complete knowledge of how, and at what rate, the system makes transitions between all states. This information generates a system of differential equations that describe the rates of change in the state probabilities. The reliability or availability of the system is the sum of the probabilities of being in those states whose definition is consistent with system success.

The above transition rates, in turn, are based on the rates of occurrence of faults, errors, detections, reconfigurations, repairs, etc. If these rates are constant with time, the process is called time-homogeneous. Many high reliability fault-tolerant systems, however, do not possess this characteristic of constant rates, and thus a time-homogeneous Markov process may not be the correct model. Markov models with time-varying parameters are called nonhomogeneous. Implicit with their use is the necessity that the transition rates be a function of global time (measured from the start of the mission) rather than local time (measured from time of entry into a particular state). This requirement precludes using this type of model for repairable systems (since repair rate transitions must be measured from the time of entry into an "undergoing repair" state), thus limiting their use for availability analyses. If the restriction to global time is removed, the result is a semi-Markov process. This type of model is considerably more difficult to solve than either of those described previously. All three types; homogeneous, non-homogeneous, and semi-Markov have been used in reliability modeling.

The Markov method approach, though flexible to accommodate analysis of a wide variety of fault tolerant designs and recovery mechanizations, has practical limitations. The very construct of system states that are comprehensive enough to represent a large number of system components and detailed enough to model the behavior of complex fault management schemes requires a very large number of system states (approaching 10^5 for highly complex systems). A common solution to this large-state space problem is to partition the system into smaller subsystems, solve each subsystem individually, and then combine the subsystem solutions to obtain the system solution. Only if the subsystem's fault tolerant behaviors are mutually independent is the system solution exact. If subsystem dependencies in fact do exist, then the assumption of independence results in an approximation.

An alternative approach that has found favor among some of the current models is to decompose the system into separate fault-occurrence and fault handling submodels. It has been observed that fault occurrence is a relatively slow process (time between faults may take days, weeks, or months) while fault handling is usually rapid (seconds or even shorter). The usual procedure is to solve the fault handling model in isolation and then aggregate the resulting effectiveness measures of the fault occurrence submodel that describe the process of redundancy depletion. The aggregated model is then solved to obtain the predicted reliability.

The more sophisticated fault-handling models currently in use can be described generally as single-entry, three-exit processes. Entry is at the occurrence of a fault. Intermediate states are defined to describe the logical progress from fault occurrence through tne steps of detection, isolation, recovery and/or reconfiguration for permanent, transient and intermittent types of faults. The three exit states can be characterized as:

- **Transient Restoration** - Correct ıecognition and handling of a transient condition

- **Covered Fault** - Correct reconfiguration of the system to handle an actual permanent fault or transient fault which is mistaken as permanent
- **System Failure** - Either the fault causes a system failure by itself (single-point failure) or a second fault occurs before the original fault is covered.

Figure 7-3 illustrates the general single fault-handling model used in the *Computer-Aided Reliability Estimator - third generation* (CARE III) developed by NASA Langley and available through COSMIC. This composite model contains the three types of faults represented in CARE III:

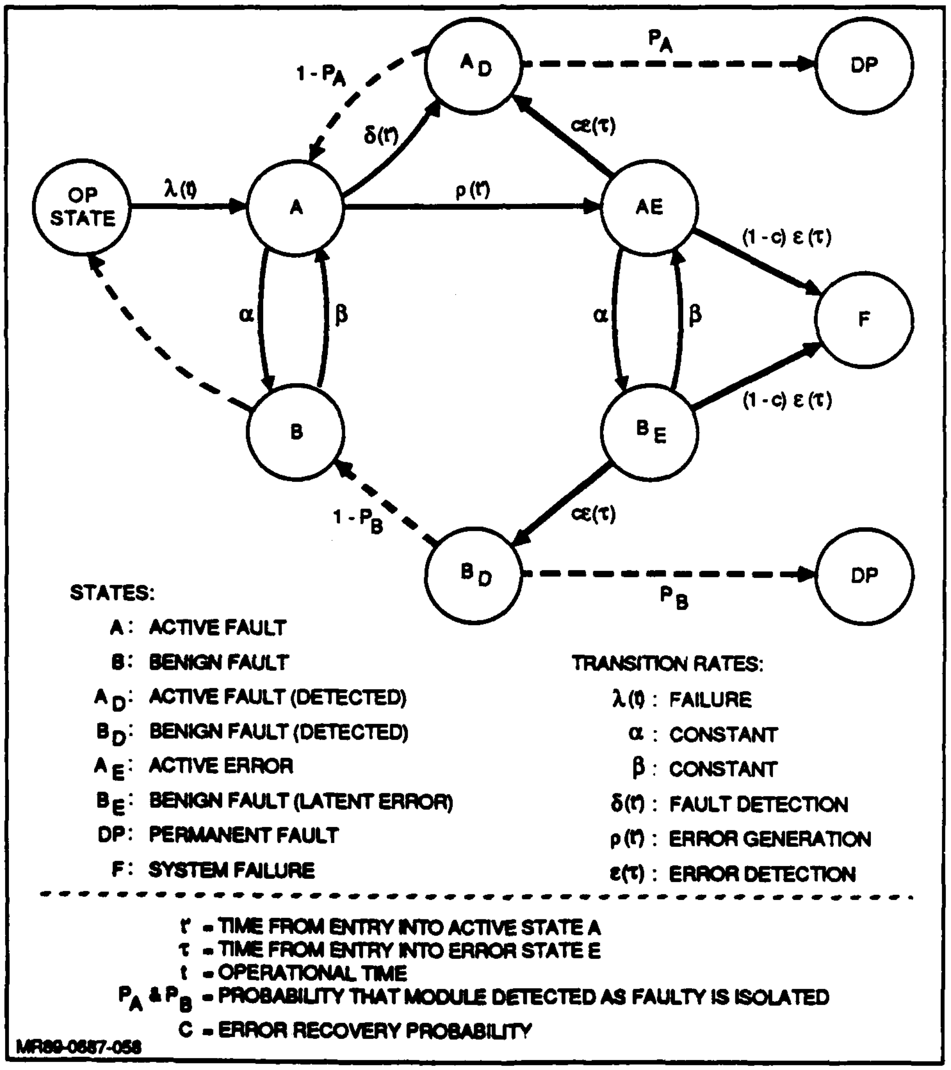


Figure 7-3. CARE III General Single Fault-Handling Model.

permanent, intermittent, and transient. The different fault types are modeled by assigning appropriate values or functions which connect the model states. One set of connecting values or functions (transition parameters) defines a model for a particular fault type.

In general, inputs for this type of fault-handling model are quite detailed and require knowledge of the distribution and parameter values of detection, isolation, recovery, rates, etc. Output consists of the exit rates or probability of exiting each of the exit states. To determine system reliability, these values are combined in the fault occurrence/ exhaustion of redundancy model.

As with most analyses, existing models of complex systems are subject to errors and constraints. They may be introduced by the model designer who, in order to obtain a solution and working within the constraints imposed by a particular model, finds it necessary to make certain assumptions or approximations. The user introduces errors whenever his model construct fails to properly represent the system under investigation. These can occur either when the relationships between constituent system elements are not described properly or when input parameters are incorrect in their characterization (distributional assumptions) or precision (value assumptions).

7.2.2 Assumptions and Limitations of Current Models

A number of the major assumptions, limitations, and sources of error present in existing reliability models are identified below:

- Solving a fault-handling model in isolation and then reflecting its results in an aggregate model is, itself, an approximation technique. The assumptions necessary to determine a solution typically result in a lower bound (conservative) approximation of the system reliability
- Separate fault-handling models have been assumed to be independent of system state. This requires that the same fault-handling model and choice of parameters be used irrespective of the system's level of degradation. This ignores the fact that for many systems the recovery process is faster if the number of active

units is smaller or that the recovery process may be different, depending on the sequence of events in different subsystems

- The common technique of partitioning the system into independent functional subgroups for computational ease is a potential source of error. The magnitude and direction of the error is a function of how truly independent/dependent the subgroups are of each other. If subgroups are assumed independent when in fact they are not, the effect is an overstatement of system reliability/availability. If subgroups are assumed completely dependent when some degree of independence exists, the effect is an understatement of the system's reliability/availability
- Some models assume a constant instantaneous fault-protection coverage factor in lieu of a separate fault handling model. These fail to recognize that time spent in the intermediate fault-handling states to detect, isolate, and recover/reconfigure are non-zero random variables during which a second item failure could result in system failure. Further, as with fault handling models, these times are generally not constant, but depend on the current state of the system
- Most models require the assumption that the system is perfect at the mission start. Therefore, they cannot evaluate the effects of latent defects (e.g., handling, manufacturing, transportation, prior mission), nor assist in determining the testability payoff or requirements for detecting and removing them before the start of the mission. Models with this limitation cannot be used to evaluate alternate maintenance concepts that include degradation between missions as an acceptable strategy
- Some models require that spares be treated exactly like active units, irrespective of their actual utilization in the system mechanization. This requires that spares are assumed to be "hot" and have the same failure rates and failure modes as the active units. This assumption will cause the model to understate the system reliability in those situations where spares are "cold" or in "standby" and/or where their failure rates may be less than those of the active units

- As indicated previously, some models require the assumption that item failure rates are constant throughout time. This will result in an overstatement of system reliability if the items have failure rates that increase with mission time. Some models remove the restriction and permit time-varying failure rates. However, the solution the algorithms employ requires the use of global time (as opposed to local time of entry into a state), thus precluding the use of the model for repairable systems and availability analysis.

It is important that the analyst be aware of these limitations so that the model chosen is the most appropriate for the system under review.

The characteristics of the *Automated Reliability Interactive Estimation System* (ARIES), CARE III, *Hybrid Automated Reliability Predictor* (HARP), and *Semi-Markov Unreliability Range Evaluator* (SURE) are summarized in Fig. 7-4. These represent a sampling of the models in existence and are a cross-section (in terms of type, application, limitation,

	MODEL			
ATTRIBUTE	ARIES	CARE III	HARP	SURE
SIZE	LARGE SYSTEMS	LARGE SYSTEMS	SMALL SYSTEMS	SMALL SYSTEMS
MATURITY	MATURE	MATURE	RELATIVELY NEW	RELATIVELY NEW
RESULT	LOWER BOUND	LOWER BOUND	LOWER BOUND	UPPER & LOWER BOUNDS
SOLUTION	EIGENVALUE	NUMERICAL INTEGRATION	NUMERICAL INTEGRATION	NEW ALGEBRAIC THEORY
MODEL TYPE	HOMOGENEOUS MARKOV	SEMI-MARKOV	NON-HOMOGENEOUS MARKOV	SEMI-MARKOV
INPUT	SYSTEM STATE	FAULT TREE	FAULT TREE OR SYSTEM STATE	SYSTEM STATE
REPAIRABLE SYSTEM	YES	NO	YES	NO
FAILURE RATES	EXPONENTIAL	EXPONENTIAL OR WEIBULL	WEIBULL FOR NON-REPAIRABLE SYSTEMS	EXPONENTIAL
FAULT HANDLING	INSTANTANEOUS CONSTANT FAULT PROTECTION COVERAGE FACTORS ARE INPUT	INCLUDES A SEPARATE MODEL WHICH IS INDEPENDENT OF SYSTEM STATE	CHOICE OF 7 MODELS, ONE OF WHICH IS A SIMULATION	NO DETAILED PARAMETRIC MODEL. INPUT SAMPLE MEANS AND VARIANCES OF RECOVERY.
SPARES	SPARES HAVE OWN FAILURE RATES	HOT WITH SAME FAILURE RATE AS ACTIVE UNITS	SPARES HAVE OWN FAILURE RATES	WHEN COLD, FAILURE RATE IS ZERO. WHEN HOT, SAME FAILURE RATE AS ACTIVE UNITS.

R88-0687-059
R88-7339-072

Figure 7-4. Characteristics of Current Reliability Models.

and state-of-the-art) of the more modern models used for reliability/availability asse:sment of fault-tolerant systems.

7.3 Operational Readiness & Availability

Two useful measures that address system capability are operational readiness and availability. Although different, some analysts confuse the distinction and treat them synonymously. Both measures are useful in discussions of system effectiveness. Both concepts relate the operating time between failures to some longer time period but differ in what is to be included in this longer time period. To differentiate between these two separate and useful concepts the following definitions are helpful:

- **Availability** - a measure of the degree to which an item is in an operable and committable state at the start of a mission when the mission is called for at an unknown (random) time. Availability calculations typically include operating time, active repair time, administrative time, and logistic time
- **Operational Readiness** - the ability of a system to respond to an operational plan upon receipt of an operations order. Total calendar time is the basis for computation of operational readiness.

Availability and operational readiness tradeoff analyses are used to evaluate the impact of a system's R/M/T design features in conjunction with operational and mission requirements. Major factors that influence these measures include:

- Reliability, maintainability, and testability design characteristics
- Field maintenance concept employed (e.g., conventional organizational, intermediate, and depot-level maintenance; or a two-level maintenance concept consisting of organizational and depot maintenance levels)
- Logistic resources available
- Mission and design characteristics.

Figure 7-5 shows these factors and other relationships affecting readiness.

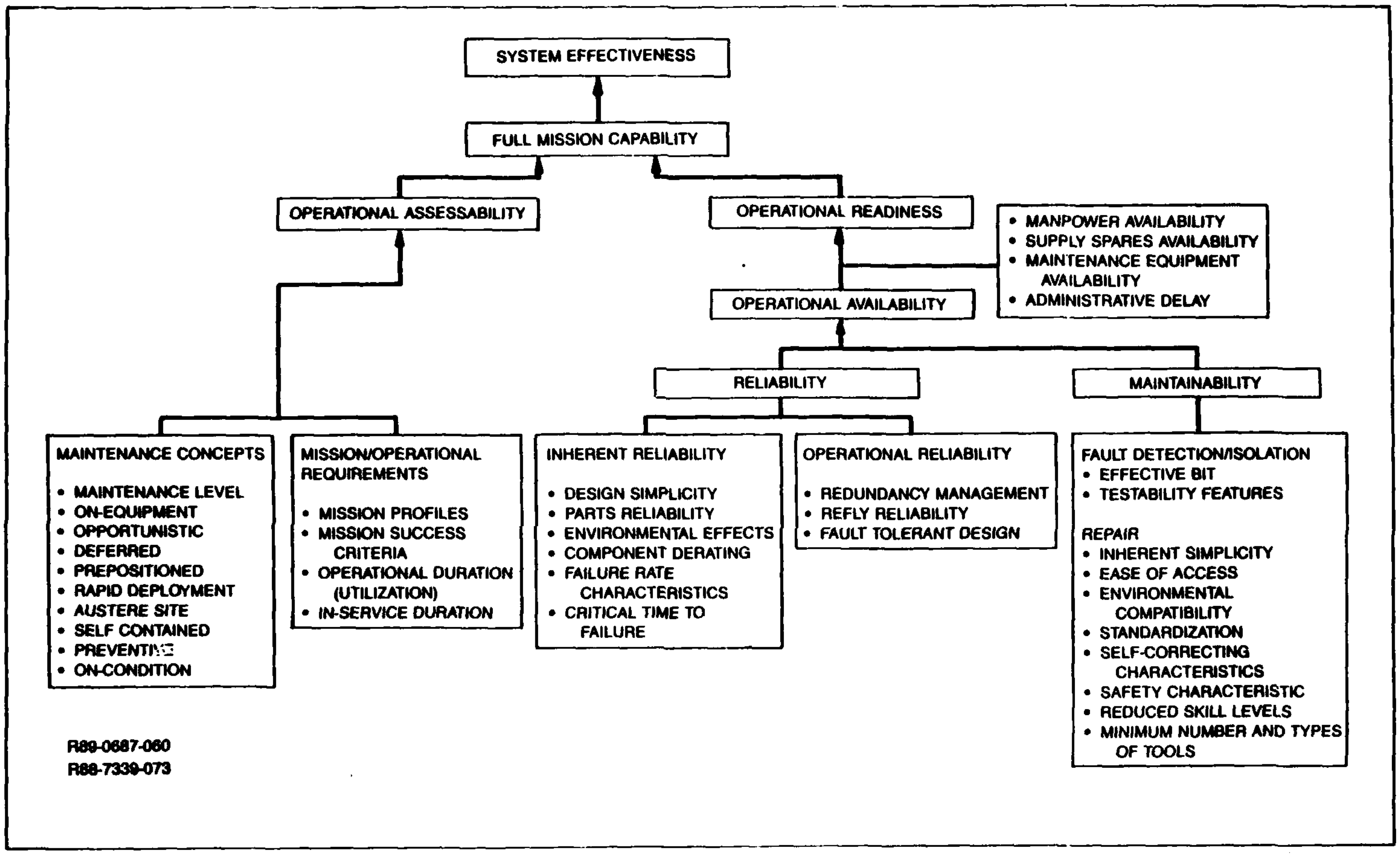


Figure 7-5. Factors & Relationships Affecting Readiness & System Effectiveness.

7.3.1 Inherent Availability

A very useful measure of system readiness that is often evaluated during conceptual and preliminary design/detailed design phases involves the classical steady-state inherent availability (A_i) relationship:

$$A_i = \frac{MTBM}{MTBM + MTTR} \tag{7-1}$$

The inherent availability of a system or equipment is the probability that it is operating satisfactorily at any point in time when used under stated conditions, where the time considered is operating time and active repair time. Thus, A_i excludes from consideration all free time, storage time, administrative time, and logistic time. As the name indicates, A_i refers primarily to the built-in capability of the system or equipment to operate satisfactorily under stated conditions.

Assessment of A_i permits realistic assignment of responsibility in the event that an unsatisfactory availability situation exists. If an improvement in A_i is indicated, responsibility can be properly assigned to the design and production engineers, assuming of course, that the operating conditions are compatible with design specifications. On the other hand, if system readiness is unsatisfactory and improvement in A_i is not indicated, then responsibility may be placed on the commander or civilian administrator to effect the required improvement by reducing administrative and logistic delays. If neither of these steps is indicated and operational readiness is unsatisfactory, improvement depends on changes in free time and storage time, implying more efficient use of the system equipment.

Clearly, A_i embodies the R/M/T system attributes that are most directly under the control of designers and technical managers. System MTBM is a direct result of equipment selection, duty cycle, operating environment(s) and fault tolerance. System MTTR reflects design decisions involving equipment FD/FI, accessibility, and installation provisions.

7.3.2 Operational Availability

The operational availability (readiness) of a system is determined principally by maintenance frequency and the "repairability" characteristics of the design, and is dependent upon the probability of system repair within a prescribed period of "downtime" for corrective and preventive maintenance. This may be expressed by the ratio:

$$\text{Operational availability } (A_o) = \frac{\text{Uptime}}{\text{Uptime} + \text{Downtime}} \qquad (7\text{-}2)$$

The relationships of "uptime" and "downtime" components are described in Fig. 7-6 and definitions are provided in Fig. 7-7. Uptime is a function of the MTBM and the downtime is a function of the mean restore time (MRT). Hence:

$$A_o = \frac{\text{MTBM}}{\text{MTBM} + \text{MRT}} \qquad (7\text{-}3)$$

$$\text{Where: } \text{MRT} = \text{MTTR} + \text{MLDT} + \text{MDT} \qquad (7\text{-}4)$$

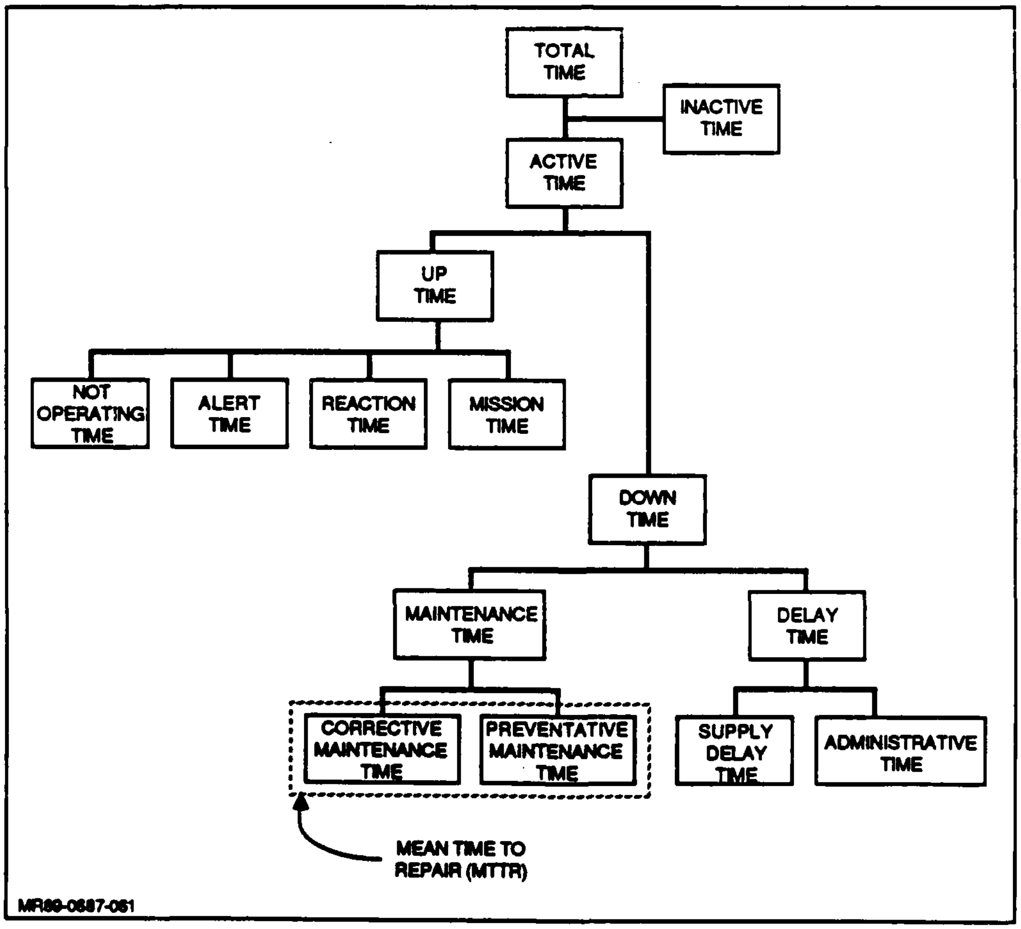


Figure 7-6. Availability Component Time Relationships.

COMPONENT	DEFINITION
ACTIVE TIME	THE ELAPSED TIME DURING WHICH AN ITEM IS IN AN OPERATIONAL INVENTORY
ADMINISTRATIVE TIME	THAT ELEMENT OF DELAY TIME NOT INCLUDED IN SUPPLY DELAY TIME
ALERT TIME	THE ELAPSED TIME DURING WHICH AN ITEM IS ASSUMED TO BE IN SPECIFIED OPERATING CONDITIONS AND IS AWAITING A COMMAND TO PERFORM ITS INTENDED MISSION
CORRECTIVE MAINTENANCE TIME	THE ELAPSED MAINTENANCE TIME DURING WHICH CORRECTIVE MAINTENANCE IS PERFORMED ON AN ITEM
DELAY TIME	THAT PART OF DOWNTIME DURING WHICH NO MAINTENANCE IS BEING ACCOMPLISHED ON THE ITEM BECAUSE OF EITHER SUPPLY OR ADMINISTRATIVE DELAY
DOWNTIME	THE ELAPSED TIME DURING WHICH AN ITEM IS NOT IN A CONDITION TO PERFORM A REQUIRED FUNCTION
INACTIVE TIME	THE ELAPSED TIME DURING WHICH AN ITEM IS IN RESERVE
MAINTENANCE TIME	THE ELAPSED TIME DURING WHICH MAINTENANCE IS PERFORMED
MISSION TIME	THE ELAPSED TIME DURING WHICH AN ITEM IS REQUIRED TO PERFORM A STATED MISSION PROFILE
NOT OPERATING TIME	THE ELAPSED TIME DURING WHICH AN ITEM IS NOT REQUIRED TO OPERATE
PREVENTIVE MAINTENANCE TIME	THE ELAPSED MAINTENANCE TIME DURING WHICH PREVENTIVE MAINTENANCE IS PERFORMED ON AN ITEM
REACTION TIME	THE ELAPSED TIME NEEDED TO INITIATE A MISSION, MEASURED FROM THE TIME COMMAND IS RECEIVED
SUPPLY DELAY TIME	THE ELAPSED DELAY TIME DURING WHICH A REPLACEMENT ITEM IS BEING OBTAINED
UPTIME R89-0687-062 R88-7339-075	THE ELAPSED TIME DURING WHICH AN ITEM IS READY TO PERFORM A REQUIRED FUNCTION

Figure 7-7. Availability Component Time Definitions.

MTBM considers the following maintenance actions:

- Functional failures (reliability sensitive)
- Scheduled maintenance (calendar/age sensitive)
- Supporting maintenance (accessibility sensitive)
- Inspections (safety sensitive)
- Cannibalizations (logistics sensitive)
- False alarms (testability sensitive).

MRT is a function of the elapsed downtime and includes:

- Mean time to repair (MTTR) - The average time to detect, isolate and repair a malfunction and restore the system to a satisfactory performance level. Included are both corrective and scheduled maintenance.
- Mean logistics delay time (MLDT) - The average time for spares to reach the system for installation.
- Maintenance downtime (MDT) - The average delay time resulting from nonproductive maintenance administration, including waiting time for facilities, test equipment, manpower, etc.

MTBM and MTTR are functions of the design, and are constants. Supply and awaiting maintenance delays are functions of logistics management and are minimized with effective management and planning. With effective managbement and supply the MRT reduces to the MTTR and equation (7-3) approaches the classical steady-state inherent availability relationship given by equation[1] (7-1). Therefore the influence of reliability, maintainability, and testability on system readiness is a function of balancing the parameter variables as a function of operational and mission need, optimizing resources, and the use of R/M/T/design techniques which serve to increase system fault tolerance and fault detection/isolation capability, decrease equipment failure rate, and improve equipment accessibility for repair or replacement.

7.3.3 Availability Design Trades

Reliability, maintainability and testability attributes of fault tolerant C^3I systems can be evaluated through design tradeoffs to achieve a balance of system supportability features with operational and mission needs, and program resources. As the maintenance frequency is decreased through improved reliability, the inherent availability of a system will approach 100%. Similarly, maintainability design improvements can reduce the number of false alarms and expedite maintenance by reducing trouble-shooting time. This improves the availability of the system by increasing the interval between maintenance (MTBM) and reducing the MTTR. The availability ratio, MTTR/MTBM, is used extensively in design tradeoffs to assess the R/M/T impact on system availability, as illustrated in Fig. 7-8. As this ratio decreases, either through an increase in the

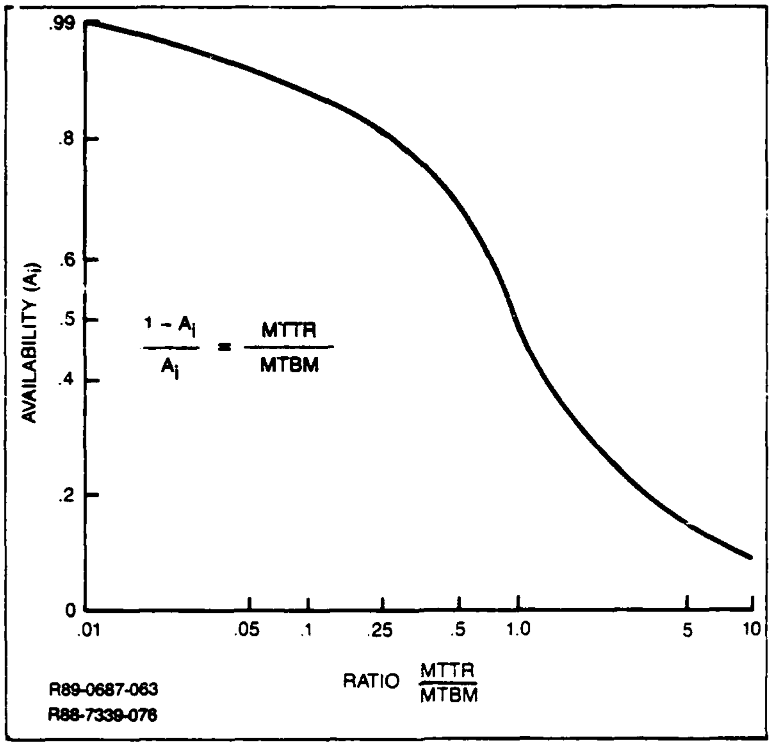


Figure 7-8. Relationship of Inherent Availability and MTTR and MTBM.

maintenance interval or reduction in the restore time, the system availability/readiness improves.

Utilization becomes a factor in systems that are susceptible to relatively long periods of inactivity and brief actual operating times; in these cases adjustments must be made to the availability calculations. Since down time can be represented only in continuous, elapsed (or calendar) time, it is convenient to introduce a factor, K, which when multiplied by the MTBM, will express the frequency or mean time rate in terms of calendar time. The expression developed is:

$$A_i = \frac{MTBM(K)}{MTBM(K) + MTTR} \qquad (7\text{-}5)$$

where:

$$K = \frac{T_c}{T_o} = \frac{\text{Calendar time per system}}{\text{Utilization per system}}$$

T_c = calendar time over which the system must be available (usually 24 hours/day x 365 days/year per system)

T_o = operating time or mission time of the system during T_c.
By definition, operating time for a system can never exceed calendar time; therefore, the reciprocal of K, 1/K, can never be greater than unity and may be expressed as an operational duty cycle which reduces equation (7-5) to:

$$\frac{1 - A_i}{A_i} = \frac{MTTR}{MTBM} \cdot \frac{1}{K} \qquad (7\text{-}6)$$

This transforms to Fig. 7-9 where utilization can be assessed as a function of various availability ratios that may result from design and configuration changes.

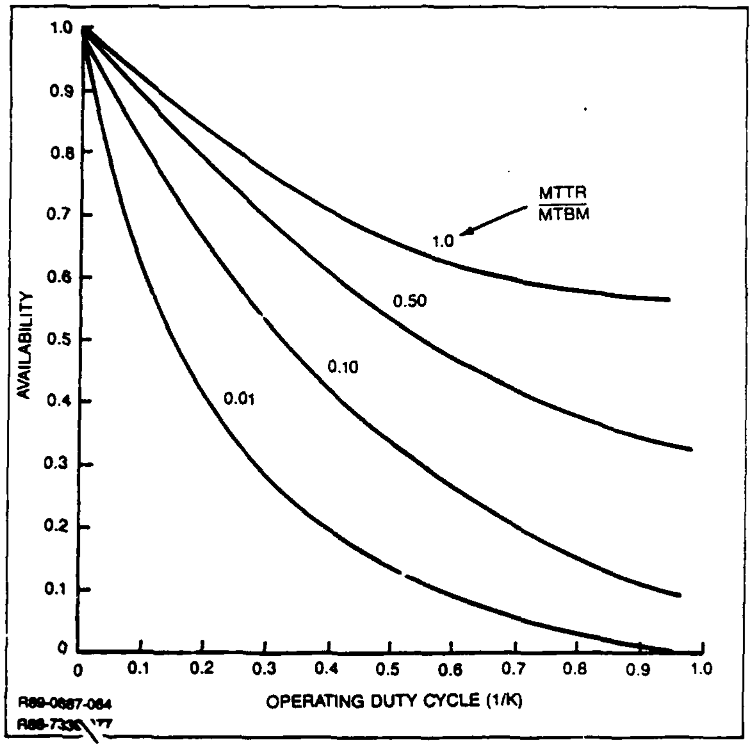


Figure 7-9. Effects of Operating Duty Cycle on Availability.

7.4 Mission Effectiveness Analysis

Mission effectiveness, E(t) is a measure of a system's capability to accomplish its mission objective within the stated operational demand time E(t) ca.. be expressed as the product of operational availability, mission reliability (R(t)), and the system pe.·formance index(P_s) as follows:

$$E(t) = A_o\ R(t)\ P_s \qquad (7\text{-}7)$$

Obviously, effectiveness is influenced by the way the system was designed and built. However, just as critical are the way the equipment is used and the way it is maintained. Hence, effectiveness can be materially influenced by the designer, the production engineer, the operator of the system, and the system's maintainer. It is also influenced by the logistic system that supports the operation, and by administrative policy decisions regarding personnel, equipment use and fiscal control.

The effectiveness expression takes into account the probability that the system will be available on operational demand (A_o), the probability of not experiencing a critical system failure ($R(t)$), and the percentage of mission objectives that can be expected to be achieved (P_s). The expression also implies that system effectiveness must be stated in terms of the requirements placed upon the system, indicating that use conditions and failure are related. As the operational stresses increase, failure frequency may also be expected to increase. If continuous operation is required, any cessation due to failure or scheduled maintenance reduces system effectiveness. If the demands of the equipment are such that an on-off use cycle provides significant free time for maintenance, system effectiveness is enhanced. Maintenance of a state of readiness on a continuous basis increases the percentage of equipments which reach an inoperable condition prior to demand for use and increase fault tolerance requirements.

For a C^3I system, the system performance index would relate the mission objectives to system capabilities such as area of surveillance, target detection probability, etc. However, it should be noted that operational requirements often exceed design objectives. For example, a decrease in target vulnerability results in a decrease in system effectiveness, and surface-to-air missiles designe [illegible] be used against subsonic aircraft are ineffective if called upon to engage supersonic targets.

System effectiveness assessment and analysis fundamentally answer three basic questions:

- Is the system working at the start of the mission (availability)?

- If the system is working at the start of the mission, will it continue to work during the mission (reliability)?
- If the system worked throughout the mission, will it achieve mission success (performance)?

R/M/T are important contributors to system effectiveness since they are significant factors in establishing system availability and dependability. However, in the total system design context, R/M/T must be integrated with other system parameters such as performance, quality, safety, human engineering, survivability/vulnerability, logistics, cost, etc, to arrive at a system configuration that optimizes effectiveness while meeting overall system requirements.

7.4.1 Optimization of System Effectiveness

The optimization of system effectiveness is important throughout the system life cycle. Effectiveness optimization is the balancing of available resources (time, money, personnel, etc) against resulting effectiveness parameters (performance, operational readiness, etc), until a combination is found that provides the most effective system for the desired expenditure of resources. Thus, the optimum system might be one that:

- Meets or exceeds a particular level of effectiveness for minimum cost, and/or
- Provides maximum effectiveness for a given total cost.

Optimization is illustrated by the flow diagram of Fig. 7-10 which shows the effectiveness optimization process as a feedback loop consisting of the following three steps:

1. Designing many systems that satisfy the operational requirements and constraints
2. Computing resultant values for effectiveness and resources used
3. Evaluating these results and making generalizations concerning appropriate combinations of design and support factors, which are then fed back into the model through the feedback loops.

7.4.2 System Effectiveness Models

A number of approaches exist to analyze system effectiveness. In particular, *Aeronautical Research Inc.* (ARINC), the *Air Force Weapon*

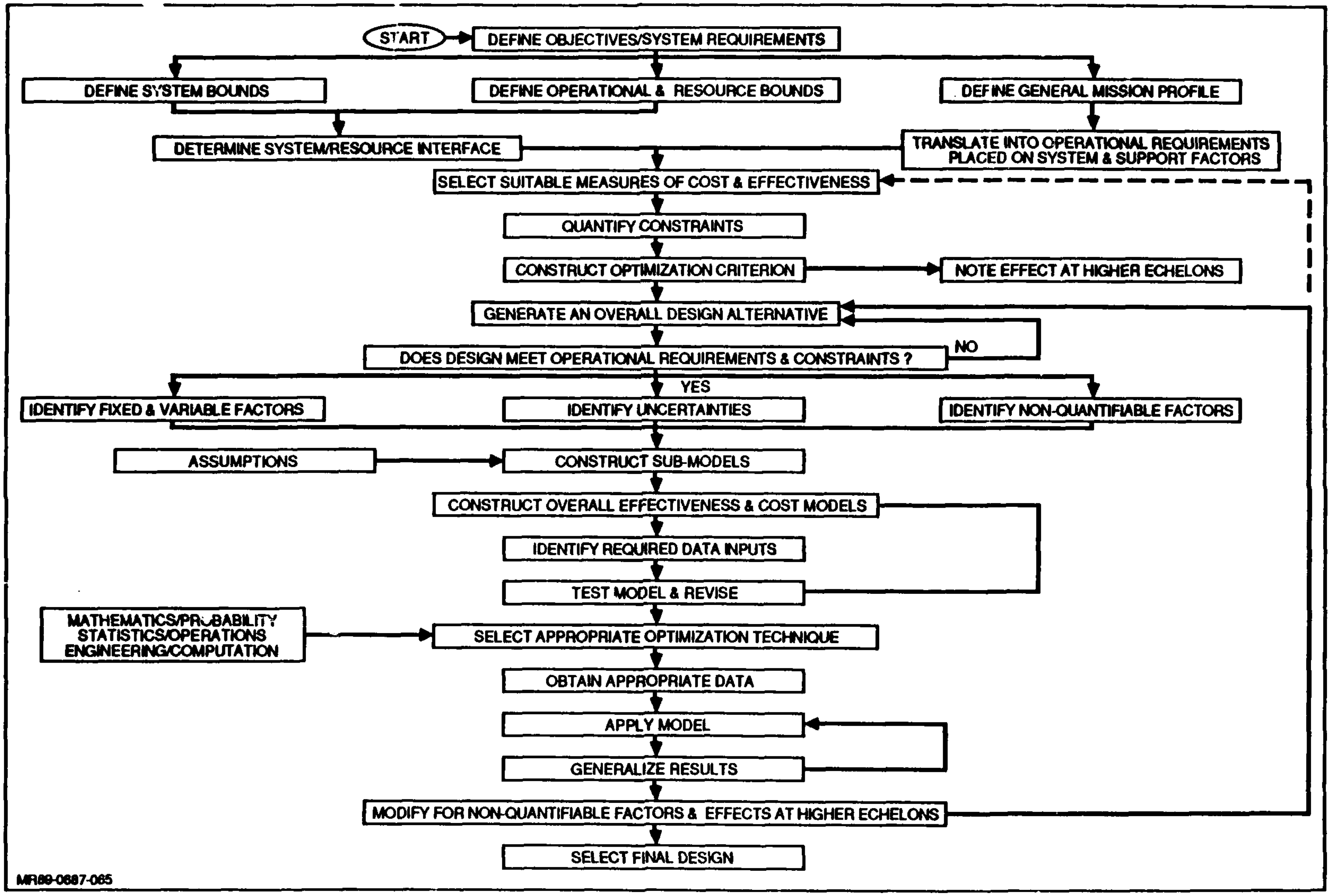

START
DEFINE OBJECTIVES/SYSTEM REQUIREMENTS
DEFINE SYSTEM BOUNDS
DEFINE OPERATIONAL & RESOURCE BOUNDS
DEFINE GENERAL MISSION PROFILE
DETERMINE SYSTEM/RESOURCE INTERFACE
TRANSLATE INTO OPERATIONAL REQUIREMENTS PLACED ON SYSTEM & SUPPORT FACTORS
SELECT SUITABLE MEASURES OF COST & EFFECTIVENESS
QUANTIFY CONSTRAINTS
CONSTRUCT OPTIMIZATION CRITERION
NOTE EFFECT AT HIGHER ECHELONS
GENERATE AN OVERALL DESIGN ALTERNATIVE
DOES DESIGN MEET OPERATIONAL REQUIREMENTS & CONSTRAINTS ?
NO
YES
IDENTIFY FIXED & VARIABLE FACTORS
IDENTIFY UNCERTAINTIES
IDENTIFY NON-QUANTIFIABLE FACTORS
ASSUMPTIONS
CONSTRUCT SUB-MODELS
CONSTRUCT OVERALL EFFECTIVENESS & COST MODELS
IDENTIFY REQUIRED DATA INPUTS
TEST MODEL & REVISE
MATHEMATICS/PROBABILITY STATISTICS/OPERATIONS ENGINEERING/COMPUTATION
SELECT APPROPRIATE OPTIMIZATION TECHNIQUE
OBTAIN APPROPRIATE DATA
APPLY MODEL
GENERALIZE RESULTS
MODIFY FOR NON-QUANTIFIABLE FACTORS & EFFECTS AT HIGHER ECHELONS
SELECT FINAL DESIGN
MR89-0687-065

Figure 7-10. Generalized Effectiveness Optimization Process Flow Diagram.

System Effectiveness Industry Advisory Committee (WSEIAC), and the Navy have developed concepts to evaluate system effectiveness. Figure 7-11 summarizes these system effectiveness models. Although there is some variation in the design parameter and attributes identified as inputs to these models, it is clear that all require significant R/M/T inputs in addition to performance and utilization parameters.

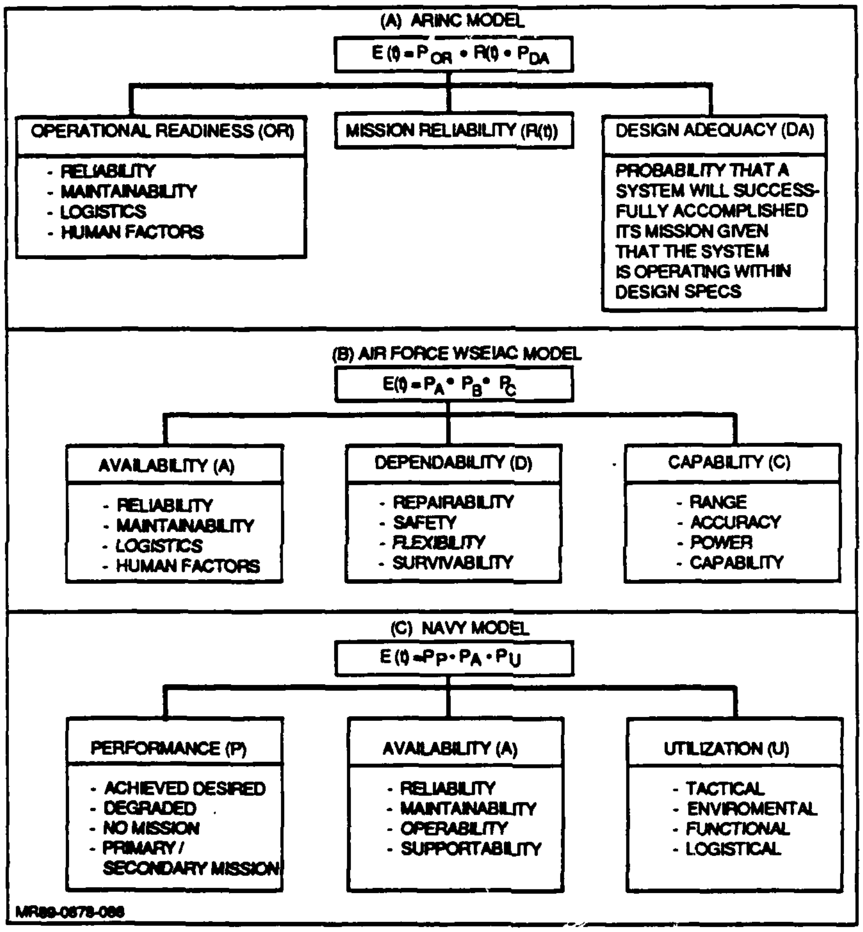


Figure 7-11. System Effectiveness Models.

7.5 Logistics Resource Analysis

To achieve a high state of readiness for a fault tolerant C³I system, consideration must be given to logistic resources (personnel, facilities, support equipment and spares). Even highly reliable fault tolerant systems, when they fail, can suffer rapid degradation in readiness if properly trained maintenance personnel, facilities, test equipment, and spares are not available. Figure 7-12 shows how a system's operational availabil-

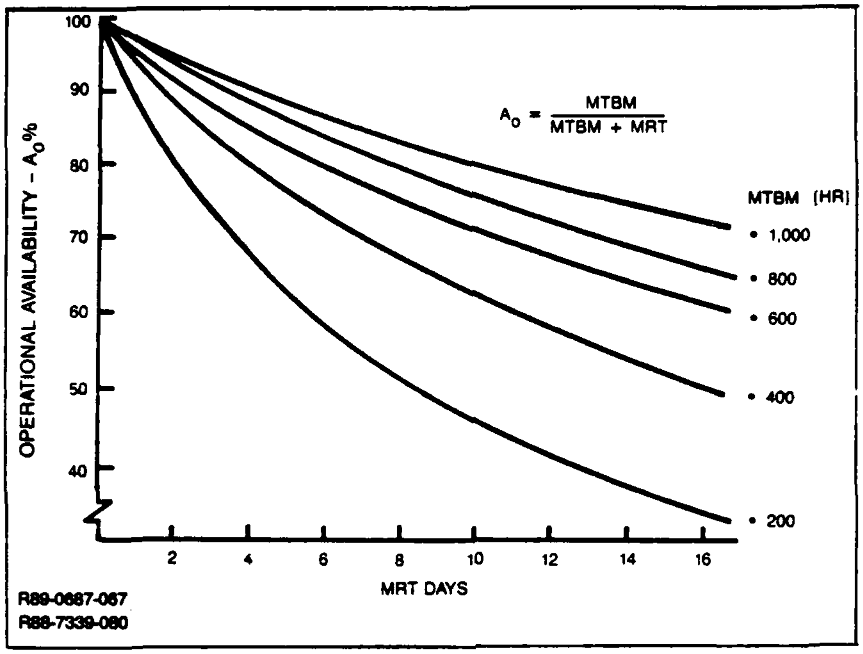


Figure 7-12. Relationships of A_o, Reliability, & MRT.

ity will decay with increasing restore time due to delays in logistic supply and paucity of maintenance personnel.

Maintenance manpower requirements are based on the number and types of skills required to perform unscheduled repairs and scheduled maintenance tasks at the anticipated maintenance frequencies. It is linked to A_o via the MDT and MTTR components of MRT and reflects both scheduled and unscheduled maintenance. Personnel manning is a function of direct unscheduled repair and depends upon the number of technicians and skill types required to perform the task and the task frequency. This relationship and the correlation to A_o are shown in Fig. 7-13. This can also be expressed by the following relationship which combines the effects of manpower loading and availability:

$$MPH/OH_i = MTTR_i \; (P_i) \; \lambda_i = \frac{MTTR_i}{MTBM_i} \cdot P_i \, , \qquad (7\text{-}8)$$

where:

MPH/OH_i = Maintenance persons hours per system operating hour

P_i = Number of technicians required to effect the repair

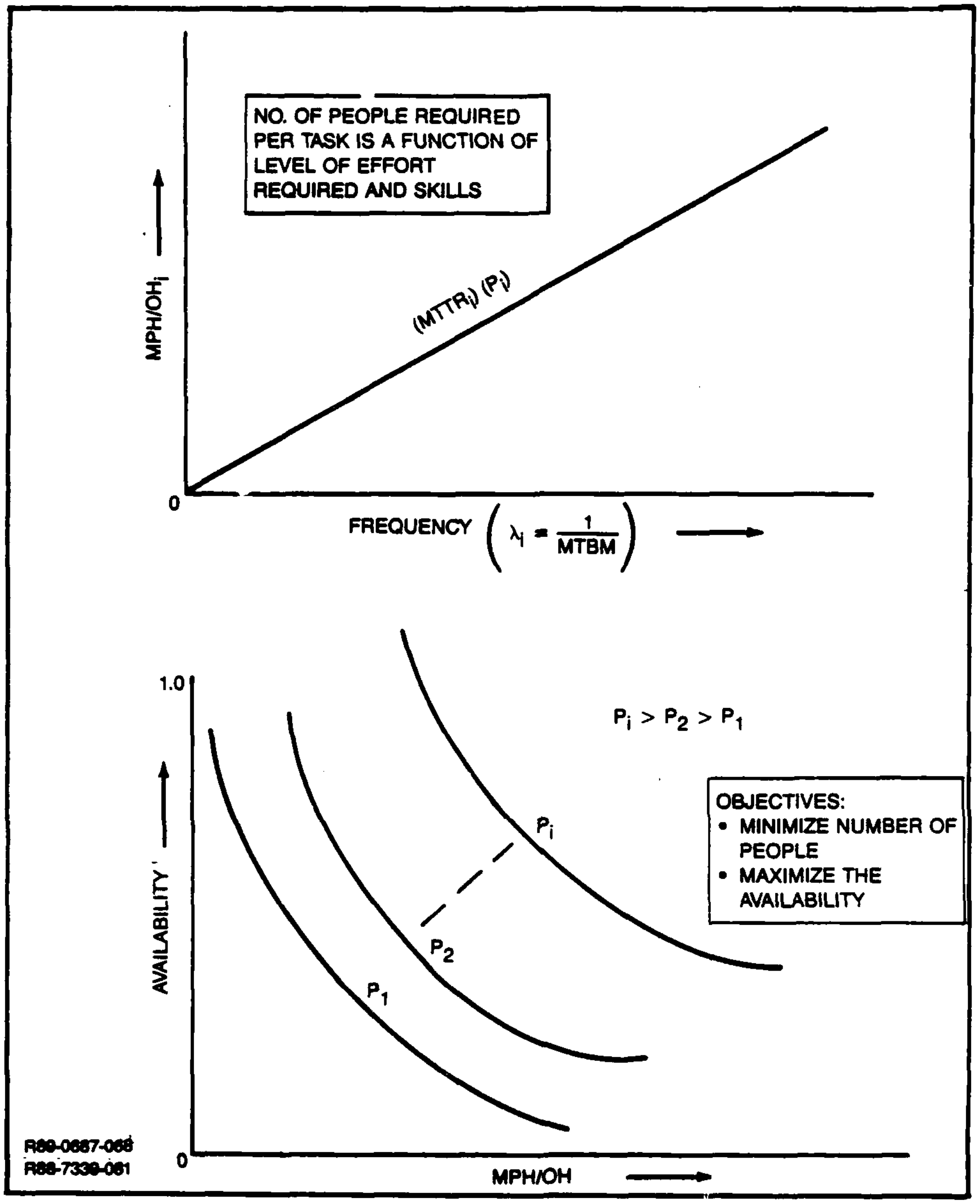


Figure 7-13. **Relationships Affecting Manpower Loading and Readiness.**

$$\lambda_i = \frac{1}{MTBM_i} \text{ rate at which the repair occurs.}$$

The MDT is a function of the following maintenance delays:

MDT_s = Mean Downtime due to scheduled maintenance

MDT_{oa} = Average downtime awaiting outside assistance

MDT_d = Average downtime due to lack of documentation

MDT_t = Average downtime due to lack of training

MDT_{or} = Average downtime due to other reasons (including unavailability of tools, facilities and support equipment)

The total delay time is expressed as the sum of these delays:

$$MDT = MDT_s + MDT_{oa} + MDT_d + MDT_t + MDT_{or} \tag{7-9}$$

The mean logistic downtime (MLDT) is the delay associated with replacing failed components with spares, and is a function of the logic in defining the level or repair (assembly, component, etc) and on what level it will be performed (on equipment, intermediate repair level, depot repair, etc). Figure 7-14 provides a typical logic model for building up logistic delay times. This results in the general expression for MLDT;

$$MLDT = K_1 T_1 + (1 - K_1) [RK_2T_2 + R(1 - K_2) T_3 + (1 - R)T_4] \tag{7-10}$$

where:

K_1 = Percent of spares available for repair of the end-item

$(1-K_1)$ = Percent of defective equipment that will be repaired by piece-part replacement

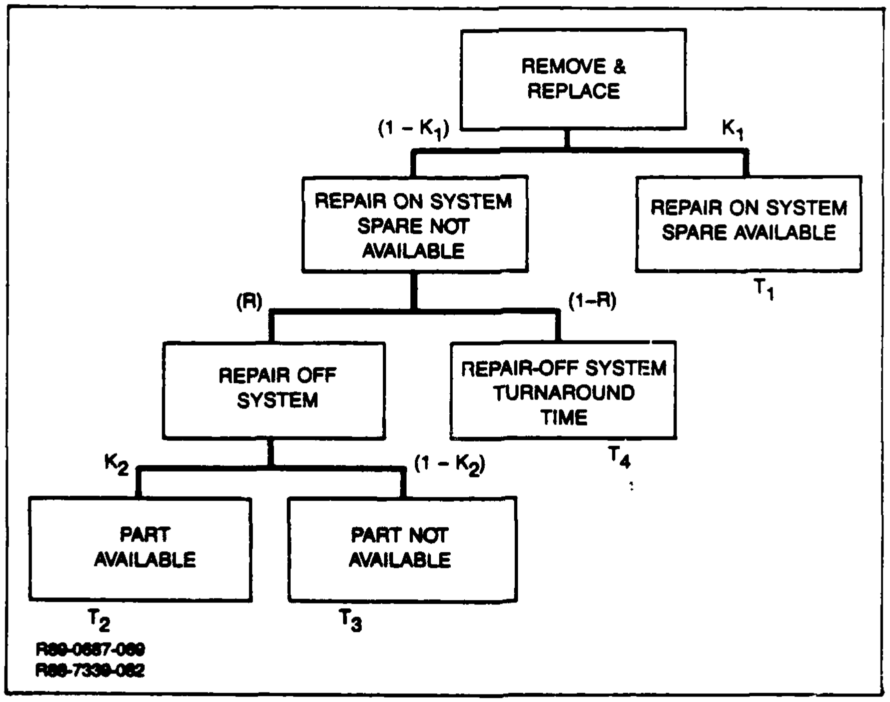


Figure 7-14. Functional Mean Logistic Downtime Model.

K_2 = Percent of piece-parts available for repair

$(1-K_2)$ = Percent of piece-parts unavailable for repair

T_1 = Time for repair of an end item by equipment removal and replacement

T_2 = Time for repair of equipments by piece-part replacement

T_3 = Delay time for repair of an equipment when piece-parts are not available and have to be requisitioned from a forward stockage point

T_4 = Repair cycle time for off-site repair of defective equipment

R = Percent of equipment repaired on-site

$1 - R$ = Percent of equipment repaired off-site.

Sparing requirements for a program normally are determined by performing a level of repair analysis as part of logistic support analysis activity. This analysis establishes the most economical level of repair (assembly, subassembly, component) and identifies where the repair should be accomplished (organization, intermediate, depot) based on the maintenance concept.

7.6 Life Cycle Cost Analysis

Design tradeoffs are not meaningful unless they can be expressed in terms of a common parameter. **LCC has been found to be the best common denominator for normalizing the effects of the diverse variables associated with R/M/T. R/M/T has a significant effect on incurred costs throughout all phases of a product's life cycle. Therefore, it is essential that the technical manager have an understanding of how this effect is manifested. The R&M levels achieved by equipment have a major impact on the support costs incurred throughout the system life cycle.** Figure 7-~~15 shows~~ that up-front R/M/T effort is mandatory during the conceptual design phases. **Up to 70% of the total LCC is committed before concept definition; and up to 85% of the operations and support costs are determined by FSD.** The reliability level determines how often an item fails, thus establishing the frequency with which maintenance resources (personnel, spares, checkout equipment, etc) are required. Maintainability characteristics of the design determine how long it takes to correct each

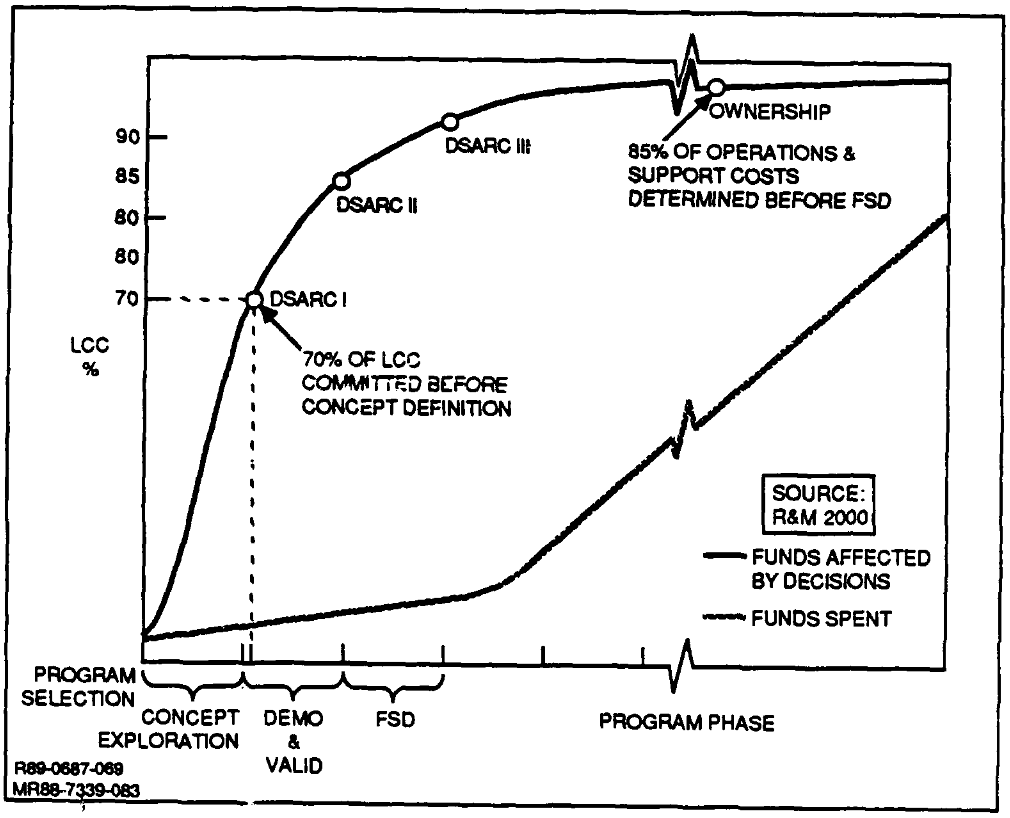


Figure 7-15. Percent of LCC Committed per Program Phase.

malfunction. The combined effect of these achieved levels reflects the total utilization, and hence the cost of the resources necessary to maintain the equipment ing its operational phase. However, these levels are essentially predetermined by the emphasis on R/M/T during the design and development phases. Therefore, the effort to minimize/control support costs must begin early in the design phase and must include deliberate actions to "design in" R/M/T.

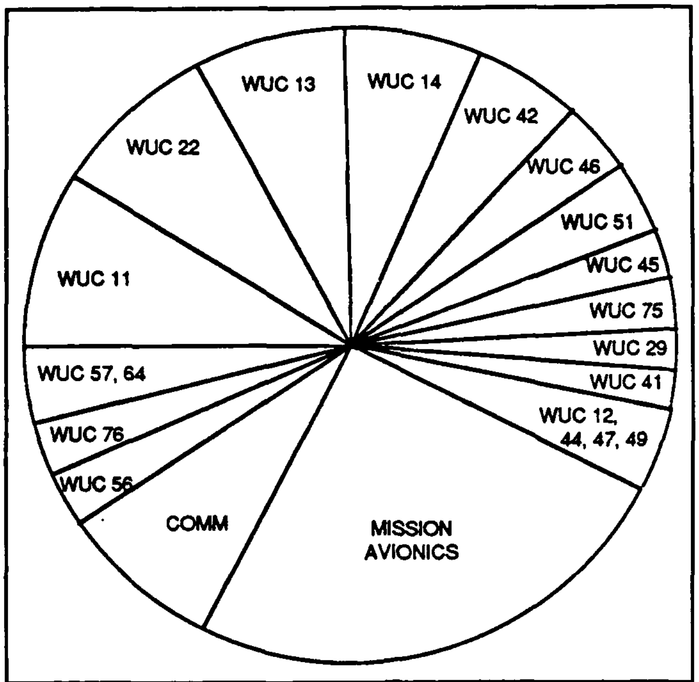
It should be noted that performance and reliability often have competing interests, and the incorporation of one may be accomplished at the expense of the other. Redundancy or environmental isolation, which increases reliability, adds weight which may detract from such C^3I system performance parameters as speed or range. Achievement of more dynamic performance usually results in lower reliability levels than those realized from more benign operating conditions. Higher reliability usually is associated with simplicity (i.e., few parts and interfaces), whereas additional performance capability is generally achieved by increased system complexity.

The benefits toward lower support costs derived from incorporation of R&M must be traded off against potentially higher acquisition costs. Furthermore, specification of R/M/T requirements must be viewed in terms of their potential impact on performance and, conversely, the effect of performance on achieved R&M levels must be understood. This is best accomplished by a design process in which performance, R/M/T, and cost are considered equally and systematically.

7.6.1 Effect of R/M/T Levels on LCC

The combined effect of achieved reliability and maintainability levels is reflected in the expenditure of resources necessary to maintain a system during its operational phase. One such resource is the personnel required to perform corrective maintenance. A parameter commonly used to measure expenditure of this resource is maintenance person hours per system operating hour (MPH/OH). This measure has been found useful for assessment and comparison because it embodies the joint influence of several key R&M factors (i.e., failure frequency, elapsed maintenance time, number of maintenance personnel), and is normalized by system utilization. Figure 7-16 shows composite distributions of MPH/OH by work unit code (WUC) for a typical aircraft. This data provides an indication of the relative contribution of each WUC so that high cost drivers may be identified. The avionics suit for this aircraft was found to require approximately 42% of the MPH/OH. C^3I system designers should pay particular attention to mission avionics system requirements and equipments in new designs since these areas might benefit markedly from R/M/T improvements and yield reduced LCC.

To support conceptual design studies and tradeoff analyses, R&M indices can be estimated if R&M data is available on similar systems. In particular, relationships that permit the estimation of MTBF, MTBM, and MPH/OH can be developed using multiple regression techniques, provided that high-level design and operational parameters are available on a large enough population of similar systems. As an example, Fig. 7-17 illus-



WUC	SUBSYSTEM DESCRIPTION	PERCENT CONTRIBUTION
11	AIRFRAME	8.8%
22	ENGINE	8.3
13	LANDING GEAR	7.6
14	FLIGHT CONTROL	7.0
42	ELECTRICAL POWER	5.3
46	FUEL SYSTEM	3.7
51	INSTRUMENTS	3.7
45	HYDRAULIC/PNEUMATIC POWER	2.6
75	WEAPON DELIVERY	2.5
29	POWER PLANT INSTALLATION	2.3
41	AIR CONDITIONING	2.0
12	FUSELAGE COMPARTMENT	1.8
44	LIGHTING	1.6
49	MISCELLANEOUS UTILITES	0.6
47	OXYGEN	0.4
	TOTAL NON-AVIONICS	58.2%
MA*	MISSION AVIONICS	25.1
COMM**	VOICE, NAV, IFF, DATA COMM	8.0
56	FLIGHT REFERENCE	2.9
76	ELECTRONIC CONTERMEASURES	2.9
57	INTEGRATED GUIDANCE	2.5
64	INTERPHONE	0.4
	TOTAL AVIONICS	41.8%

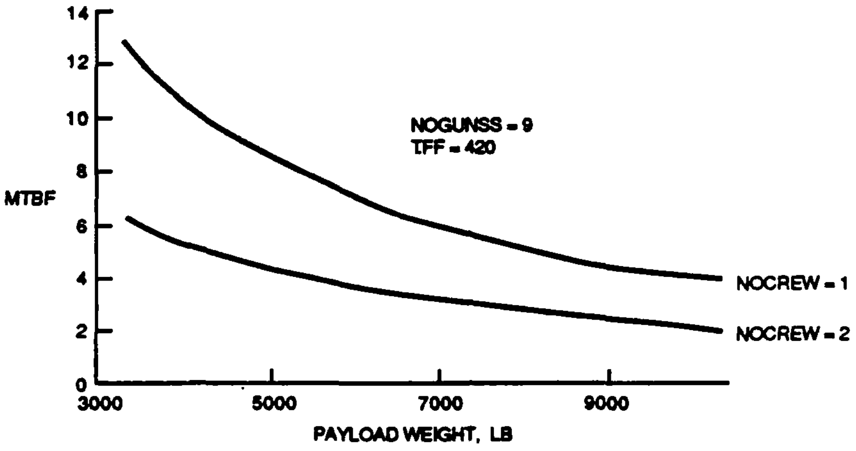
* MISSION AVIONICS INCLUDES WUC 72,73 & 74

* * COMM INCLUDES WUC 61,62,63,65,66,67,69,& 71

MR89-0687-070

Figure 7-16. Distribution of Corrective Manhours for Typical Aircraft.

$$MTBF = 1.0448 \times 10^{+3} (NOCREW)^{-0.9897} (NOGUNSS)^{-0.9855} (PYLDWT)^{-1.1277} (TFF)^{+1.1536}$$



NOCREW – NUMBER OF CREW **DATA BASE RANGE: 1 TO 2**

...INDICATES AIRCRAFT SIZE AND COMPLEXITY OF THE ONBOARD SYSTEMS. A LARGER AIRCRAFT WITH INCREASED SYSTEM COMPLEXITY WILL TEND TO EXPERIENCE MORE EQUIPMENT FAILURES

NOGUNSS – NUMBER OF GUNS PLUS STORE STATIONS **DATA BASE RANGE: 5 TO 10**

...MEASURES THE MISSION AVIONICS COMPLEXITY, WHICH SHOULD BE DIRECTLY PROPORTIONAL TO THE NUMBER OF AVIONIC EQUIPMENT FAILURES

PYLDWT – PAYLOAD WEIGHT **DATA BASE RANGE: 3600 TO 10296 LB**

...MEASURES AVIONIC COMPLEXITY. INCREASED COMPLEXITY TENDS TO INCREASE THE NUMBER OF EQUIPMENT FAILURES

TFF – TIME OF FIRST FLIGHT **DATA BASE RANGE: 96 TO 259 MONTHS SINCE 1950**

...MEASURES TECHNOLOGY ADVANCEMENT AND CAPABILITY OF THE EQUIPMENT COMPONENTS. ELECTRONIC AND STRUCTURAL IMPROVEMENTS HAVE RAISED THE RELIABILITY OF AIRCRAFT COMPONENTS

MR89-0687-071

Figure 7-17. Typical Aircraft Total Avionic System MTBF Estimating Relationship.

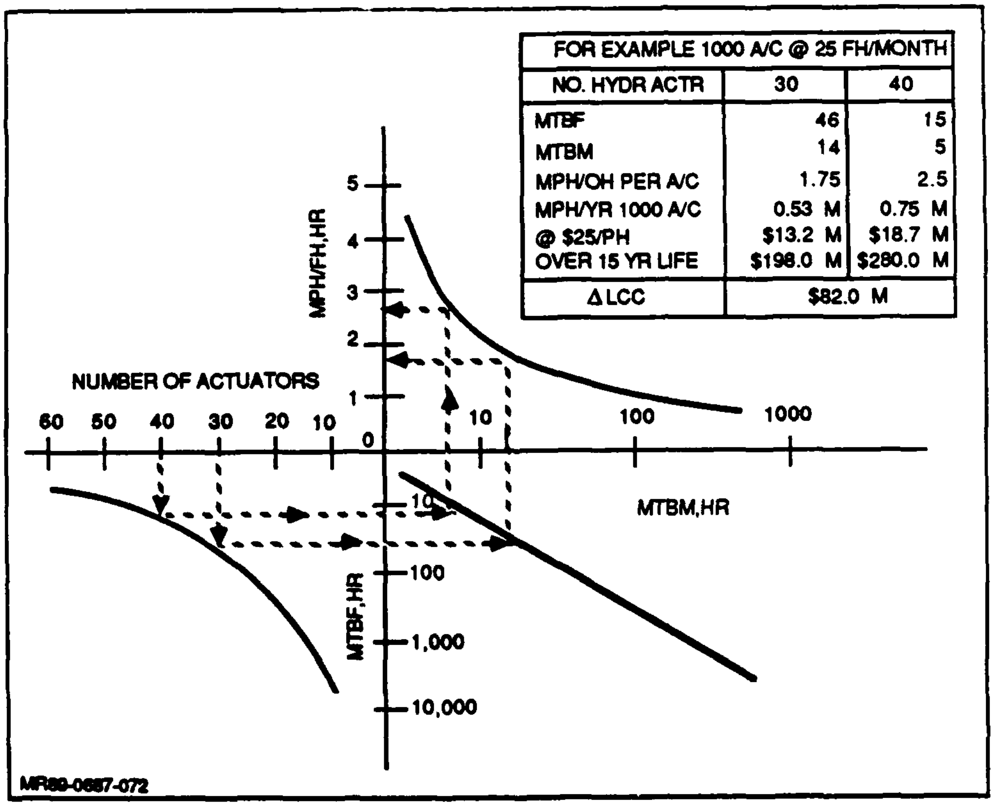
trates an estimating relationship for the MTBF of the total avionic system. This relationship provides an MTBF estimate as a function of payload weight, number of guns plus store stations, number of crew, and a technology factor given by time of first flight. Similar relationships can be developed for MTBM and MPH/OH. The MTBF and MTBM values calculated using these relationships can be compared to "target" or "goal" values as part of conceptual design tradeoff studies. The MPH/OH values can be coupled with assumptions concerning:

- Number of systems in the program
- Utilization rate
- Labor rate

to estimate the maintenance personnel cost component of LCC. For example:

- Operating hours/year = number of systems x operating hours/system/month x (12 months/year)
- Maintenance person hours/year = operating hours/year x maintenance person hours/operating hour
- Maintenance personnel cost/year = maintenance person hours/year x personnel cost/hour.

In addition to using relationships of this type separately to estimate a particular R&M parameter, nomographs can be developed for particular programs by proper juxtaposition of the curves and provide a convenient and direct method of performing LCC tradeoffs. Nomographs of this type permit the user to directly read the resulting estimated values of MTBF, MTBM, and MPH/OH by starting with performance alternatives and drawing lines from curve to curve. Figure 7-18 illustrates this technique and



FOR EXAMPLE 1000 A/C @ 25 FH/MONTH		
NO. HYDR ACTR	30	40
MTBF	46	15
MTBM	14	5
MPH/OH PER A/C	1.75	2.5
MPH/YR 1000 A/C	0.53 M	0.75 M
@ $25/PH	$13.2 M	$18.7 M
OVER 15 YR LIFE	$198.0 M	$280.0 M
ΔLCC	$82.0 M	

Figure 7-18. High Level LCC Tradeoff - Number of Aircraft FCS Actuators vs MPH/OH.

shows the impact of flight control complexity on other personnel costs associated with maintaining the flight control system (FCS).

7.6.2 Cost-Benefit Analysis for Fault Tolerance

The principal reason for incorporating fault tolerant design features in C^3I systems is to improve reliability and reduce system downtime. A convenient way to determine the degree of fault tolerance warranted in the design, or the levels of redundancy required, would be to perform a cost-benefit analysis on the design. However, although one can measure the costs associated with the acquisition of a system and its operation and maintenance (personnel, support equipment, spares, technical publications, etc), the cost of having a multimillion-dollar system unavailable for a mission is difficult to measure. In view of this it is often more convenient to think in terms of the value of a ready-hour for the system. If the problem is addressed from a total-force-level approach, it can be seen that a small quantity of systems with high-ready rates can be as effective as a larger number of systems with lower-ready rates. The real question, then, is to identify the breakpoint, i.e., the point at which it is more cost-effective to procure additional systems than to incorporate readiness improvements.

To this end it is sometimes necessary to work with "worth" rather than "cost" directly, since costs can be converted to the worth of a ready hour by dividing the anticipate life-cycle cost of the system by the number of ready hours (requirement or goal) during the system's life-cycle. As shown in the following equation:

$$RI = \frac{\text{LCC Per System}}{\text{RR x SL x 365 Days/Yr x 24 Hrs/Day}} = \frac{\text{Total Dollars}}{\text{Ready-Hours}} \qquad (7\text{-}11)$$

where:

RI = Readiness index (worth of a ready-hour)

RR= Readiness rate

SL = Service life (years)

Using this criteria, any R/M/T improvement to the system can be evaluated for cost effectiveness. As an example, for a system with a readiness goal of 80%, a service life of 20 years, and an anticipated LCC of $75,000,000 per system, a readiness index of $535/ready-hour is obtained. For this particular system, if the cost of saving one ready-hour over its service life exceeds $535, the R/M/T improvement should not be implemented.

7.6.3 Life Cycle Cost Models

A number of LCC models have been developed by industry and government to estimate cost and provide relationships between significant and controllable acquisition and operations/support costs. Many of the models are basically accounting structures which contain terms and factors for each cost element of a system life cycle. Other models contain relationships between two or more cost factors and may contain cost-estimating relationships for cost elements which cannot easily be determined until the system is committed to field use. One model in particular provides R&M estimates for fighter/attack and cargo/transport aircraft in addition to LCC. The *Air Force Modular Life Cycle Cost Model* (MLCCM) contains cost estimating relationships for all phases of the system life cycle. In addition, MLCCM provides R&M estimates (MTBF, MTBM, MPH/OH) based upon aircraft high-level physical characteristics. These R&M estimates can be compared to program goals or requirements, and the impact on LCC of changes in R&M numerical requirements can be evaluated.

7.7 R/M/T Evaluation & Tradeoff Analysis Checklist Questions

The following checklist questions are useful to determine whether appropriate analyses have been conducted to assure that a C^{3}I system design meets applicable R/M/T and availability requirements:

a. Have the results of readiness analysis been used for:

- **Support of design trades?**
- **Optimization of support systems?**
- **Identification of readiness risks?**

b. Have qualitative and quantitative R&M requirements been met? (e.g., probability of success, MTBF, MTTR, MPH/FH, A_o, FO/FS, etc)

c. Has an FMEA (that reflects the latest system design) been performed to identify all single point failures?

d. Has a reliability model of the system (including all redundant elements, and redundancy control devices) been developed and analyzed?

e. Have all system interfaces been identified and analyzed?

f. Have computerized models capable of analyzing fault tolerant systems been used in the reliability analysis?

g. Does the reliability evaluation model that was used in the analysis properly account for the effect of imperfect fault coverage?

h. Has the system reliability, maintainability, and availability been accurately modeled to reflect the benefits of fault tolerance?

i. Are the overall fault tolerance provisions that were incorporated in the system too extensive? Could they be reduced to save program cost without jeopardizing mission goals?

j. How credible is the reliability/availability model and supporting input data?

k. Are the system partitioning, subsystem interfaces and fault isolation mechanisms at subsystem boundaries described clearly and adequately so as to cover the given fault assumptions (i.e., fault types and classes)?

l. Does each subsystem contain sufficient error detection, fault diagnostic, and recovery provisions?

m. Are the costs associated with the fault tolerance provisions consistent with the system performance requirements?

n. If required by the system specification, can the system execute concurrent recoveries in two or more subsystems?

o. Has the occurrence of unexpected faults (although unlikely) been treated as a possible catastrophic event?

p. Have simulations been conducted to verify subsystem interactions and to test recovery algorithms?

APPENDIX A
ACRONYMS

AF	Air Force
A_i	Inherent Availability
A_o	Operational Availability
AI	Artificial Intelligence
ARIES	Automated Reliability Interactive Estimation System (Computer Program)
ARINC	Aeronautical Research Incorporated
ATE	Automatic Test Equipment
AWACS	Airborne Warning & Control System
BCH	Base-Chaudhuri Hoequeghem
BIT	Built-In Test
BSC	Binary Synchronous Communication
CARE	Computer Aided Reliability Estimation (Computer Program)
C^3I	Command, Control, Communications and Intelligence
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CND	Cannot Duplicate
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit
CSMA/CD	Carrier Sense Multiple Access/Collision Detect
D	Depot
DOD	Department of Defense
DSP	Defense Support Program
DT&E	Development, Test and Evaluation
ECC	Error Correction Codes
ECP	Engineering Change Proposal
ESS	Environmental Stress Screening

FCS	Flight Control System
FD	Fault Detection
FFI	Fraction of Faults Isolatable
FI	Fault Isolation
FMEA	Failure Mode Effects Analysis
FMECA	Failure Mode, Effects and Criticality Analysis
FO	Fail Operational
FO/FS	Fail Operational/Fail-Safe
FO^2/FS	Fail Operational/Fail Operational/Fail-Safe
FRACAS	Failure Reporting, Analysis and Corrective Action System
FS	Fail Safe
FSD	Full-Scale Development (Phase)
GFE	Government Furnished Equipment
GSE	Ground Support Equipment
HARP	Hybrid Automated Reliability Predictor (Computer Program)
I	Intermediate
IIRA	Integrated Inertial Reference Assembly
ILS	Integrated Logistic Support
I/O	Input/Output
IR	Infrared
JTIDS	Joint Tactical Information Distribution System
LCC	Life-Cycle Cost
LRM	Line Replaceable Module
LRU	Line Replaceable Unit
LSA	Logistic Support Analysis
LSI	Large Scale Integration
LTPB	Linear Token-Passing Data Bus
MDT	Maintenance Downtime
MLCCM	Modular Life Cycle Cost Model
MLDT	Mean Logistics Delay Time

MPH/OH	Maintenance Person Hours Per System Operating Hour
MRT	Mean Repair Time
MTBCF	Mission-Time-Between-Critical-Failure
MTBF	Mean-Time-Between-Failure
MTBMA	Mean-Time-Between-Maintenance-Action
MTBMI	Mean-Time-Between-Maintenance-Inherent
MTTR	Mean-Time-To-Repair
NMR	N-Modular Redundancy
O	Organizational
O&S	Operating and Support
OTH-B	Over the Horizon-Backscatter
PDR	Preliminary Design Review
R_m	Probability of Mission Success
R&M	Reliability and Maintainability
RAM	Random Access Memory
RFP	Request For Proposal
RIW	Reliability Improvement Warranty
R/M/T	Reliability, Maintainability, Testability
RQT	Reliability Qualification Test
RT	Remote Terminal
RVT	Reliability Verification Test
SIFT	Software Implemented Fault Tolerance
SOW	Statement of Work
SRR	System Requirements Review
SRU	Shop Replaceable Unit
SURE	Semi-Markov Unreliability Range Evaluator (Computer Program)
TMR	Triple Modular Redundancy
TPS	Test Program Set

VHSIC	Very High Speed Integrated Circuit
VLSI	Very Large Scale Integration
WRA	Weapon Replaceable Assembly
WSEIAC	Weapon System Effectiveness Industry Advisory Committee
WUC	Work Unit Code

APPENDIX B

GLOSSARY OF RELIABILITY, MAINTAINABILITY, TESTABILITY AND FAULT TOLERANCE TERMS

AVAILABILITY: A measure of the degree to which an item is in an operable and committable state at the start of a mission when the mission is called for at unknown (random) time. (Item state at start of a mission includes the combined effects of readiness-related system reliability and maintainability parameters, but excludes mission time.) (Ref 24)

COMMON MODE FAILURE: A non-random event, which is usually time or stress dependent, which is caused by a latent manufacturing defect, a design flaw, or a susceptibility to an unanticipated environment.

COVERAGE, FAULT PROTECTION: The conditional probability that the system will recover should a fault occur. The specification of the types of errors against which a particular redundancy scheme guards. (Ref 3)

DEPENDABILITY: A measure of the degree to which an item is operable and capable of performing its required function at any (random) time during a specified mission profile, given item availability at the start of the mission. (Item state during a mission includes the combined effects of reliability and maintainability parameters but excludes non-mission time.) (Ref 24)

DERATING: The operation of an item at less severe stresses than those for which it is rated.

ENVIRONMENTAL STRESS SCREENING: A test or series of tests, specifically designed to disclose weak parts or workmanship defects.

ERROR: An undesired resource state that exists either at the boundary or at an internal point in the resource and may be perceived as a failure when it is propagated to, and manifested at, the boundary. (Ref 15)

FAILURE: The event, or inoperable state, in which any item or part of an item does not, or would not, perform as previously specified (Ref. 24). A loss of service that is perceived by the user at the boundary of the resource. (Ref 15)

FAULT: The immediate cause of failure (e.g., maladjustment, misalignment, defect, etc) (Ref 24). The identified or hypothesized cause of the error or failure (Ref 15). A fault may be latent and undetected until it propagates and causes an error or functional failure at a higher level of operation.

FAULT, DESIGN: A generic fault designed into a function, including hardware and software faults and faults of other logical entities, such as data bus interfaces.

FAULT AVOIDANCE: Techniques which serve to prevent, by construction, the occurrence of a fault.

FAULT DETECTION: The process of determining that an error caused by a fault has occurred within the system. An undiscovered fault is classified as a latent fault.

FAULT, INTERMITTENT: Hardware faults which result in recurring inconsistent functional behavior of the hardware followed by recovery of its ability to perform within specified limits without any remedial action. Intermittent faults cannot occur in software or logic.

FAULT ISOLATION: The process of determining the location of a fault to the extent necessary to effect repair correction, or restoration to specified performance. (Ref 24)

FAULT, LATENT: A fault which exists but has not been detected.

FAULT, PERMANENT: A fault which, once it occurs, is irreversible except for permanent removal from the system.

FAULT RECOVERY: The ability of the system to provide the required service or performance or to correct errors after a fault has been detected.

FAULT, TRANSIENT: A fault not caused by a permanent defect but rather one which manifests a faulty behavior for some finite time and then is fault free. A permanent or intermittent fault which only occasionally produces discrepant results is not a transient fault.

FAULT TOLERANCE: A survivable attribute of a system that allows it to deliver its expected service after faults have manifested themselves within the system. (Ref 15)

FAULT TOLERANT SYSTEM: A system that has provisions to avoid failure after faults have caused errors within the system. (Ref 15)

GRACEFUL DEGRADATION: A design technique which utilizes extra hardware as part of a systems normal operating resources to assure that an acceptable performance level can be maintained.

ITEM: A generic term which may represent a system, subsystem, equipment, assembly, subassembly, etc. depending on its designation in each task. (Ref 25)

MAINTAINABILITY: The measure of the ability of an item to be retained in or restored to a specified condition when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance and repair. (Ref 24)

MEAN-TIME-BETWEEN-FAILURE (MTBF): A basic measure of the system reliability parameter related to availability and readiness. The total number of system life units, divided by the total number of events in which the system becomes unavailable to initiate its mission(s), during a stated period of time. (Ref 24)

MISSION EFFECTIVENESS: A measure of a systems capability to accomplish its mission objective within the stated operational demand time.

MISSION-TIME-BETWEEN-CRITICAL-FAILURES (MTBCF): A measure of MISSION RELIABILITY: The total amount of mission time divided by the total number of critical failures. (Ref 24)

N-VERSION PROGRAMMING: The independent generation of 2 or more function-ally equivalent programs from the same initial specification.

OPERABLE: The state of being able to perform the intended function. (Ref 24)

PROGRAM TAILORING: The process by which individual requirements are evaluated to determine suitability for a particular system development and acquisition.

REDUNDANCY: The existence of more than one means of accomplishing a given function. Each means of accomplishing the function need not necessarily be identical. (Ref 24)

REDUNDANCY, ACTIVE: The redundancy wherein all redundant items are operating simultaneously. (Ref 24)

REDUNDANCY, DYNAMIC: The implementation of redundant elements in such a way that they may be rearranged (either automatically or manually) to provide continued operation of a function.

REDUNDANCY, HYBRID: A combination of N-modular design practices with those that implement backup sparing.

REDUNDANCY, STANDBY: That redundancy wherein the alternative means of performing the function is not operating until it is activated upon failure of the primary means of performing the function.

RELIABILITY:

(a) The duration or probability of failure-free performance under stated conditions. (Ref 24)

(b) The probability that an item can perform its intended function for a specified interval under stated conditions. (For non-redundant items this is equivalent to definition (a). For redundant items this is equivalent to the definition of mission reliability.) (Ref 24)

RELIABILITY, MISSION: The ability of an item to perform its required functions for the duration of the specified mission profile (Ref 24).

STATE: A unique description of the operational status of the system, usually characterized in terms of the number of remaining (unfailed) constituent components.

TESTABILITY: A design characteristic which allows the status (operable, inoperable, or degraded) of an item to be determined and the isolation of faults within the item to be performed in a timely manner. (Ref 25)

APPENDIX C
LIST OF GOVERNMENT DOCUMENTS

AFLCP 800-39	Built-In-Test Design Guide
AFSC DH-1-9	Design Handbook - Maintainability
DOD Directive 5000.40	Reliability and Maintainability
DOD-STD-2167	Defense System Software Development
MIL-F-9490D	Flight Control System - Design Installation and Test of Piloted Aircraft, General Specification for
MIL-HDBK-217D	Reliability Prediction of Electronic Equipment
MIL-HDBK-338	Electronic Reliability Design Handbook
MIL-STD-470A	Maintainability Program for Systems and Equipment
MIL-STD-471A	Maintainability Verification/Demonstration/ Evaluation
MIL-STD-721C	Definition of Terms for Reliability and Maintainability
MIL-STD-756B	Reliability Modeling and Prediction
MIL-STD-781C	Reliability Design Qualification and Production Acceptance Tests: Exponential Distribution
MIL-STD-785B	Reliability Program for Systems and Equipment Development and Production
MIL-STD-882B	System Safety Program Requirements
MIL-STD-1388	Logistics Support Analysis
MIL-STD-1521B	Technical Reviews and Audits for Systems, Equipment, and Computer Software
MIL-STD-1629	Procedures for Performing a Failure Mode, Effects and Criticality Analysis
MIL-STD-2164(EC)	Environmental Stress Screening Process for Electronic Equipment

MIL-STD-2165	Testability Program for Electronic Systems and Equipment
OPNAVINST 3000.12	Operational Availability Handbook

LIST OF REFERENCES

1. MIL-HDBK-338, *Electronic Reliability Design Handbook.*
2. Veklerov, E. "Reliability of Redundant Systems with Unreliable Switches." IEEE Transactions on Reliability, October 1987.
3. Siewiorek, D.P., Swarz, R.S. *The Theory and Practice of Reliable System Design.* Digital Press, 1982.
4. *Defense Electronics,* August 1987.
5. Dantowitz, A., Hirschberger, G., Pravidlo, D. *Analysis of Aero nautical Equipment Environmental Failures.* Technical Report AFFDL-TR-71-32, Air Force Systems Command, Wright-Paterson Air Force Base, Ohio, May 1971.
6. Kube, F., Hirschberger, G. "An Investigation to Determine - Effective Equipment Environmental Acceptance Test Methods." ADR 14-04-73.2, Grumman Aerospace Corporation, April 1973.
7. RADC-TR-87-225. "Improved Operational Readiness Through Environmental Stress Screening (ESS)." August 1987.
8. Fink, D.G., Chistiansen, D. ed. *Electronic Engineers' Handbook.* McGraw Hill Book Co., N.Y., 1982.
9. AMCP 706-124, *Engineering Design Handbook: Reliable Military Electronics.* Headquarters U.S. Army Material Command, Va.
10. AFWAL-TR-87-1028, "Advanced System Avionics (ASA) Design/ Development Program." June 1987.
11. MIL-HDBK-251, *Reliability/Design Thermal Applications.*
12. Laprie, J., Ariat, J., Beounes, C., Kanoun, K., Hourtolle, C. "Hardware and Software Fault Tolerance: Definition and Analysis of Architectural Solutions." IEEE 1987.
13. Nelson, V., Carroll, B. *Tutorial: Fault Tolerant Computing.* IEEE Computer Society Press, 1986.
14. Koren, I., Pradhan, D. "Yield and Performance Enhancement Through Redundancy in VLSI and WSI Multiprocessor Systems." IEEE Proceedings, Vol 74, No. 5, May 1986.

15. Avizienis, A., Laprie, J.C. "Dependable Computing: From Concepts to Design Diversity," *Proceedings of the IEEE,* Vol. 74, No. 5, May 1986.

16. Watterson, J. *Very High Speed Integrated Circuit Self-Test/Fault-Tolerant Workshop.* Research Triangle Institute (AD B069200L). November 1982.

17. RADC-TR-86-241, Albert, J., Partridge, M., Spillman, R. *Built-in-Test Verification Techniques.* February 1987.

18. National Security Industrial Association (NSIA), Integrated Diagnostic Group. "Guidelines for Preparation of Diagnostic Requirements." July 1986.

19. Jager, R. "Integrated Diagnostics: Extension of Testability." *1986 Proceedings of Annual Reliability and Maintainability Symposium,* 1986.

20. RADC-TR-85-148, Haller, K., Andersen, K., Zbytniewski, J., Bagnally, L. *Smart BIT.* August 1986.

21. Musson, T., "System R&M Parameters from DoDD 5000.40." Proceedings of Annual Reliability and Maintainability Symposium, 1981.

22. *AGARD Lecture Series No. 81.* 1976.

23. Hardy, C. "Impact of Reliability Improvement Warranty on Avionic Reliability." AGARD Lecture Series No. 81, 1976.

24. MIL-STD-721C, *Definition of Terms for Reliability and Maintainability.*

25. MIL-STD-2165, *Testability Program for Electronic Systems and Equipment.*

26. Murn, S. *R/M/T Design for Fault Tolerance C^3I System Survey.* Grumman Aircraft Systems Technical Report No. RMS-88-R-03. May 1988.

27. MIL-STD-1547 (USAF), *Parts, Materials, and Processes for Space and Launch Vehicles, Technical Requirements For.*

28. AFSCP 800-27, *Part Derating Guidelines.*

29. AS 4613 (Navy), *Application and Derating Requirements for Elec tronic Components.*

30. TE000-AB-GTP-010, *Parts Application and Reliability Information Manual for Navy Electronic Equipment.*

31. MIL-STD-2174 (Navy), *Derating and Application Requirements for Electronic and Electrical Parts.*

32. Conroe, D., Murn, S. RADC-TR-88-69 (Volume I), *R/M/T Design for Fault Tolerance - Program Manager's Guide*. December 1987.

33. OPNAVINST 3000.12, Operational Availability Handbook.

34. McConnel, S., Siewiorek, D. *"Synchronization and Voting,"* IEEE Transactions on Computing, Feb. 1981.

35. Caroli, J. A., *et al.*, "Reliability Demonstration Technique for Fault Tolerant Systems." *Proceedings of Annual Reliability and Maintainability Symposium*, 1987.

36. Department of the Army, Navy, and Air Force, NAVMATP 9405/DARCOM P34-1/AFLCP 800-30/AFSCP 800-30/NAVMC 2721; 1981, "Joint DARCOM/NMC/AFLC/AFSC/Commanders Built-In-Test Design Guide."

37. RADC-TR-83-316, "Hardware/Software Tradeoffs for Test Systems," 1983.

38. RADC-TR-85-150, "A Rationale and Approach for Defining and Structuring Testability Requirements," 1985.

39. RADC-TR-86-195, "Tools for Integrated Diagnostics," 1986.

40. RADC-TR-84-203, "Artificial Intelligence Application to Testability," 1984.

NOTE: Although this report references the following limited documents, no limited information has been extracted.

Item 16: USGO agencies & their contractors; specific authority: 23 Nov 82. Other requests Cdr, Naval Electronics Systems Command, ELEX, Code 61V, Wash DC 20360.

Item 20: USGO agencies and their contractors; critical technology; Aug 85. Other requests RADC (RBET) Griffiss AFB NY 13441-5700.

Item 37: USGO agencies and their contractors; administrative/operational use; Feb 84. Other requests RADC (RBET) Griffiss AFB NY 13441-5700.

Item 39: USGO agencies and their contractors; administrative/operational use; Dec 86. Other requests RADC (RBRA) Griffiss AFB NY 13441-5700.